UNIVERSIDAD DE MURCIA

ESCUELA INTERNACIONAL DE DOCTORADO

TESIS DOCTORAL

Distributed Technologies in Identity Management: An Approach to Enhancing Security and Privacy

Tecnologías de identidad distribuidas: Un enfoque para mejorar la seguridad y la privacidad

D. Rafael Torres Moreno

2024

UNIVERSIDAD DE MURCIA

ESCUELA INTERNACIONAL DE DOCTORADO

TESIS DOCTORAL

Distributed Technologies in Identity Management: An Approach to Enhancing Security and Privacy

Tecnologías de identidad distribuidas: Un enfoque para mejorar la seguridad y la privacidad

Autor: D. Rafael Torres Moreno

Directores: Dr. Antonio F. Skarmeta Gómez y
Dr. Jorge Bernal Bernabé

UNIVERSIDAD DE MURCIA

DECLARACIÓN DE AUTORÍA Y ORIGINALIDAD
DE LA TESIS PRESENTADA PARA OBTENER EL TÍTULO DE DOCTOR
Aprobado por la Comisión General de Doctorado el 19-10-2022

D./Dña. Rafael Torres Moreno

doctorando del Programa de Doctorado en

Programa de doctorado en Informática

de la Escuela Internacional de Doctorado de la Universidad Murcia, como autor/a de la tesis presentada para la obtención del título de Doctor y titulada:

Tecnologías de identidad distribuidas: Un enfoque para mejorar la seguridad y la privacidad // Distributed Technologies in Identity Management: An Approach to Enhancing Security and Privacy

y dirigida por,

D./Dña. Antonio F. Skarmeta Gómez

D./Dña. Jorge Bernal Bernábé

D./Dña.

DECLARO QUE:

La tesis es una obra original que no infringe los derechos de propiedad intelectual ni los derechos de propiedad industrial u otros, de acuerdo con el ordenamiento jurídico vigente, en particular, la Ley de Propiedad Intelectual (R.D. legislativo 1/1996, de 12 de abril, por el que se aprueba el texto refundido de la Ley de Propiedad Intelectual, modificado por la Ley 2/2019, de 1 de marzo, regularizando, aclarando y armonizando las disposiciones legales vigentes sobre la materia), en particular, las disposiciones referidas al derecho de cita, cuando se han utilizado sus resultados o publicaciones.

Si la tesis hubiera sido autorizada como tesis por compendio de publicaciones o incluyese 1 o 2 publicaciones (como prevé el artículo 29.8 del reglamento), declarar que cuenta con:

- *La aceptación por escrito de los coautores de las publicaciones de que el doctorando las presente como parte de la tesis.*
- *En su caso, la renuncia por escrito de los coautores no doctores de dichos trabajos a presentarlos como parte de otras tesis doctorales en la Universidad de Murcia o en cualquier otra universidad.*

Del mismo modo, asumo ante la Universidad cualquier responsabilidad que pudiera derivarse de la autoría o falta de originalidad del contenido de la tesis presentada, en caso de plagio, de conformidad con el ordenamiento jurídico vigente.

En Murcia, a 12 de Septiembre de 2024

Fdo.: Rafael Torres Moreno

TORRES MORENO RAFAEL - 48657396E

Firmado digitalmente por TORRES MORENO RAFAEL - 48657396E
Fecha: 2024.09.13 17:30:06 +02'00'

Esta DECLARACIÓN DE AUTORÍA Y ORIGINALIDAD debe ser insertada en la primera página de la tesis presentada para la obtención del título de Doctor.

Información básica sobre protección de sus datos personales aportados	
Responsable:	Universidad de Murcia. Avenida teniente Flomesta, 5. Edificio de la Convalecencia. 30003; Murcia. Delegado de Protección de Datos: dpd@um.es
Legitimación:	La Universidad de Murcia se encuentra legitimada para el tratamiento de sus datos por ser necesario para el cumplimiento de una obligación legal aplicable al responsable del tratamiento. art. 6.1.c) del Reglamento General de Protección de Datos
Finalidad:	Gestionar su declaración de autoría y originalidad
Destinatarios:	No se prevén comunicaciones de datos
Derechos:	Los interesados pueden ejercer sus derechos de acceso, rectificación, cancelación, oposición, limitación del tratamiento, olvido y portabilidad a través del procedimiento establecido a tal efecto en el Registro Electrónico o mediante la presentación de la correspondiente solicitud en las Oficinas de Asistencia en Materia de Registro de la Universidad de Murcia

A mi mujer, Luna y a mis padres.

No te rindas que la vida es eso,
continuar el viaje,
perseguir tus sueños,
destrabar el tiempo,
correr los escombros y
destapar el cielo.

Mario Benedetti

Agradecimientos

Quisiera expresar mi más sincero agradecimiento a mis directores de tesis, Antonio y Jorge, no sólo por permitirme desarrollar un este trabajo de investigación sino también por el apoyo y la paciencia que han demostrado estos años. Gracias por permitirme ser parte de un equipo tan espectacular. Su dedicación y esfuerzo en los proyectos de investigación internacionales han sido una fuente de inspiración y aprendizaje constante. Estoy enormemente agradecido por su tiempo, sus directrices y sus consejo. El trabajo realizado y contenido en esta tesis es una prueba irrefutable y extensa de su incansable labor y excelencia profesional.

Doy gracias a mis padres, porque vuestro ejemplo siempre me ha empujado a levantarme una y mil veces. Porque sigo siendo aquel crío que jugaba (no siempre acertadamente) a desmontar cualquier cacharro que cayese en mis manos. Al final jugar a los "marcianos"se ha convertido, además de un hobby, en mi profesión así que podemos decir esto era un camino prácticamente inevitable empezando desde Calasparra, donde descubrí aquel Prince of Persia en disquetes (todo un reto) hasta ahora, que sigo jugando aunque con equipos algo más complejos.

Agradecer a mi pareja, Luna, la paciencia y confianza depositada en mí incluso cuando ni yo mismo creía que esto fuera posible. Gracias por acompañarme en este proceso que ha sido increible pero también extremadamente exigente en lo psicológico. Gracias por no rendirte y por evitar que yo lo hiciera.

También quisiera agradecer a todos mis amigos y compañeros de Dibulibu, T3, Gaia ... Gracias a vosotros, al ambiente que generáis todo este proceso se convierte en algo mucho más ameno, divertido y enriquecedor. En especial, me gustaría dar las gracias a Jesús y Agustín, que además de dos personas maravillosísimas han estado al pie del cañón en toda circusntancia, sacando adelante el trabajo en toda circunstancia.

No en vano, quiero agradecer a todas las personas que en mi día a día me rodean por haberme acompañado en este árduo proceso que no siempre me hacía ser la mejor compañía. A todos los que me han sacado una sonrisa pese a todo y contra todo.

Finalmente, agradecer mis profesores en esta facultad su guía durante toda la etapa universitaria e investigadora. Especialmente a Antonio Skármeta, por permitirme iniciarme en este mundo pero, también, a Pedro Miguel, Gabi, Rafa, Antonio Ruiz y Fran Ros, a quienes debo el mantener con vida ese gusanillo por aprender algo más, algo distinto y a ponerlo en práctica. Gracias por todo.

Agradezco a todos aquellos que han contribuido a mi viaje, pues sin su apoyo, mis logros serían solo sueños no realizados.

Isaac Asimov

Contents

List of Figures

List of Tables

Abbreviations

AAA	Authentication, Authorization and Accounting
ACID	Atomicity, Consistency, Isolation and Durability
AP	Attribute Provider
BFT	Byzantine fault tolerance
CSR	Certificate signing request
DAG	Directed acyclic graph
DB	Database
DCI	Distributed Credential Issuance
DDoS	Denial of Service attack
DIP	Distributed Identity Provider
DLT	Distributed Ledger Technologies
dP-ABC	Distributed Private Attribute Based Credential
DPV	Distributed Password Verification
DTG	Distributed Token Generation
GDPR	General Data Protection Regulation
IdM	Identity Management
IdP	Identity Provider
IoT	Internet of Things
JSON	JavaScript Object Notation
JWKS	JSON Web Key Set
JWT	JSON web token
OIDC	OpenID connect
OTP	One-time password
P-ABC	Private Attribute Based Credential
PASTA	PAssword-based Threshold Authentication
PDP	Policy Decision Point
pIdP	Partial Identity Provider
PIN	Personal Identification Number
PoC	Proof of Concept
PRF	Pseudorandom Function
OPRF	Oblivious Pseudorandom Function
RP	Relying Party
SaaS	Software as a Service

SAML	Simple Assertion Markup Language
SP	Service Provider
vIdP	Virtual Identity Provider
XACML	eXtensible Access Control Markup Language
XML	Extensible Markup Language
SLA	Service level agreement

Resumen

La era de la información se caracterizada por una rápida expansión digital y una conectividad global sin precedentes, el concepto de identidad digital ha adquirido una relevancia crucial. La identidad digital se refiere a la representación en línea de una persona, la cual se ha convertido en un componente esencial de nuestra vida cotidiana, tanto en el ámbito personal como profesional. Esta representación digital no solo abarca datos personales básicos, sino también información más compleja que puede incluir hábitos de navegación, preferencias, interacciones sociales y transacciones financieras.

El contexto actual está definido por una creciente dependencia de plataformas digitales para realizar actividades diarias, desde la comunicación hasta las transacciones financieras y la gestión de servicios gubernamentales. Sin embargo, esta transformación también ha introducido desafíos significativos en la gestión y protección de la identidad digital. Los modelos tradicionales de gestión de identidad, que se basan en estructuras centralizadas, han demostrado ser inadecuados para enfrentar los desafíos emergentes. Estas estructuras centralizadas, en las que un único proveedor o entidad controla y almacena la identidad digital del usuario, se enfrentan a problemas de escalabilidad y seguridad, exacerbados por el ritmo acelerado de innovación tecnológica y el incremento en las preocupaciones sobre privacidad.

La gestión de identidad digital tradicional, que históricamente ha funcionado bajo un esquema de confianza centralizada, se encuentra en un punto crítico. Estos sistemas, aunque fundamentales en la infraestructura digital actual, han evolucionado de manera lenta y no han logrado adaptarse adecuadamente a los crecientes requisitos de privacidad y seguridad. A medida que los usuarios se vuelven más conscientes de los riesgos asociados con la exposición de sus datos personales y la forma en que estos datos son utilizados, las fallas inherentes en los sistemas centralizados se hacen más evidentes.

Una de las principales debilidades de estos sistemas es la concentración de confianza en una única entidad, lo que aumenta el riesgo de ataques y violaciones de seguridad. Este modelo centralizado no solo agrava los problemas de seguridad, sino que también puede llevar a la explotación indebida de datos sensibles. La falta de herramientas adecuadas para permitir a los usuarios gestionar de forma efectiva su privacidad, junto con la explotación lucrativa y a menudo invasiva de datos personales, ha generado una creciente preocupación sobre la seguridad de la identidad digital.

En este contexto, el crecimiento exponencial de los servicios en línea y la expansión del Internet de las Cosas (IoT) han introducido nuevos desafíos y oportunidades. El IoT, con su capacidad para conectar una amplia gama de dispositivos y recopilar datos

en tiempo real, plantea preguntas cruciales sobre la privacidad y la seguridad. Los sistemas actuales de gestión de identidad deben evolucionar para enfrentar estos desafíos y proporcionar soluciones innovadoras que garanticen una protección efectiva en un entorno hiperconectado.

Las tecnologías descentralizadas, como Blockchain y otras tecnologías de libro mayor distribuido (DLT, por sus siglas en inglés), emergen como respuestas prometedoras a los problemas asociados con los sistemas centralizados. Estas tecnologías ofrecen una estructura en la que la confianza se distribuye entre múltiples nodos en lugar de depender de un único punto central. Esta descentralización tiene el potencial de abordar muchas de las deficiencias de los sistemas tradicionales, ofreciendo mejoras en términos de seguridad, transparencia y control sobre los datos personales. No obstante, estas tecnologías aún enfrentan desafíos significativos, incluyendo problemas de privacidad, la gestión de claves y la interoperabilidad con estándares existentes.

Los sistemas de identidad autosoberana (SSI, por sus siglas en inglés) representan una evolución crucial en la gestión de identidad. Estos sistemas permiten a los usuarios mantener el control total sobre sus datos personales, reduciendo la dependencia de proveedores centralizados y, por ende, los riesgos asociados con ellos. Los sistemas SSI promueven un modelo en el que los usuarios pueden gestionar, compartir y verificar su identidad de manera segura y eficiente, sin la intervención de intermediarios que podrían comprometer su privacidad.

Esta tesis se enfoca en explorar la evolución hacia un modelo de gestión de identidad descentralizado, que aproveche las tecnologías de privacidad y DLT para ofrecer una solución más segura, robusta y confiable. El objetivo es mitigar los riesgos inherentes a los sistemas actuales, tales como el fraude y la suplantación de identidad, y mejorar la confianza en la infraestructura tecnológica mediante un enfoque basado en la descentralización y la autosuficiencia.

En suma, esta investigación se centra en cómo las innovaciones tecnológicas pueden transformar la gestión de identidad digital, abordando las deficiencias de los sistemas existentes y estableciendo un nuevo estándar en términos de seguridad, privacidad y confianza.

Motivación

La necesidad de una evolución en los sistemas de gestión de identidad es impulsada por el crecimiento continuo de los servicios en línea y la expansión del Internet de las Cosas (IoT). Los sistemas tradicionales de IdM, como los basados en X.509 y Single Sign-On (SSO), han demostrado ser inadecuados para enfrentar los desafíos modernos de privacidad, seguridad y confianza. Estos sistemas, aunque ampliamente utilizados, presentan limitaciones significativas que afectan la protección de los datos personales y la capacidad de los usuarios para controlar su identidad digital.

Uno de los problemas principales con los sistemas tradicionales es la centralización de la confianza en un único proveedor de identidad. Este modelo centralizado no solo

aumenta el riesgo de ataques y violaciones de seguridad, sino que también deja los datos personales expuestos a un control y potencial mal uso por parte de una sola entidad. Además, la falta de herramientas efectivas para proteger la privacidad de los usuarios y evitar el rastreo por parte de los proveedores de identidad es una preocupación creciente.

Las tecnologías emergentes basadas en Blockchain y DLT ofrecen un enfoque alternativo que distribuye la confianza entre múltiples nodos en lugar de centralizarla. Este enfoque descentralizado tiene el potencial de abordar muchas de las deficiencias de los sistemas tradicionales, al reducir el riesgo de ataques únicos y mejorar la transparencia. Sin embargo, estas tecnologías aún no han alcanzado un nivel de madurez que permita su adopción generalizada como soluciones completas para la gestión de identidad.

El principal desafío es construir un ecosistema robusto y completo que pueda integrar estas tecnologías de manera efectiva. A pesar de los avances en la implementación de soluciones como Shocard y Serto, estos sistemas todavía enfrentan problemas relacionados con la gestión de claves, la privacidad y la interoperabilidad con estándares existentes. Para que un sistema de gestión de identidad descentralizado sea viable, debe cumplir con una serie de requisitos clave que aseguren la protección de los datos personales, la privacidad del usuario y la confianza en la infraestructura tecnológica.

Entre los requisitos principales para un sistema moderno de gestión de identidad se encuentran:

R1 *Minimización de datos*: Es crucial procesar y almacenar los datos personales de manera adecuada y limitada, evitando la recopilación y retención innecesaria de información.

R2 *Prevención del rastreo*: El sistema debe prevenir que los proveedores de identidad rastreen y monitoreen a los usuarios a través de sus actividades en línea.

R3 *Mejora de la confianza*: Se debe permitir una verificación confiable de las identidades dentro de la infraestructura existente, garantizando que las identidades digitales sean auténticas y verificables.

R4 *Minimización de hardware*: La solución debe evitar la necesidad de hardware específico para su adopción generalizada, permitiendo una implementación amplia y accesible.

R5 *Integración con estándares existentes*: El sistema debe ser compatible con tecnologías y estándares existentes como OpenID, OAuth y SAML, para facilitar su integración con soluciones preexistentes.

R6 *Transparencia*: Debe informar claramente sobre cómo se protegen los datos personales y cómo se utilizan las tecnologías para garantizar la privacidad.

R7 *Usabilidad*: La experiencia de usuario debe ser comparable a las soluciones más adoptadas actualmente, asegurando que la implementación de nuevas tecnologías no afecte negativamente la facilidad de uso.

R8 *Cumplimiento normativo*: Es fundamental que el sistema cumpla con las regulaciones de protección de datos, como el Reglamento General de Protección de Datos (GDPR), para garantizar el respeto a los derechos de los usuarios.

Estos requisitos buscan crear un sistema de gestión de identidad que equilibre privacidad, seguridad y funcionalidad, satisfaciendo las demandas de usuarios, proveedores y reguladores. La implementación efectiva de estos requisitos es esencial para desarrollar una solución que pueda abordar las deficiencias de los sistemas actuales y proporcionar una gestión de identidad digital robusta y confiable.

Objetivos y Metodología

El análisis de las soluciones actuales para la gestión de identidad revela tanto logros significativos como áreas críticas que requieren mejoras. Aunque los enfoques actuales han logrado avances notables, también presentan deficiencias y oportunidades de mejora en términos de privacidad, confianza y seguridad. Estas deficiencias están particularmente relacionadas con los requisitos R1, R2, R3 y R8, los cuales ofrecen un área rica para investigaciones y mejoras adicionales.

El objetivo principal de esta tesis es abordar estas deficiencias y proporcionar una solución integral para la gestión de identidad digital que incorpore características avanzadas de privacidad, confianza y seguridad. El objetivo se puede expresar de la siguiente manera:

> *Analizar, diseñar y validar soluciones para la gestión de identidad digital que incluyan características avanzadas de privacidad, confianza y seguridad, manteniendo o mejorando los niveles de seguridad de las soluciones existentes, sin perjudicar la usabilidad y alineadas con las regulaciones de protección de datos personales.*

El objetivo prioritario es ofrecer un sistema integral de gestión de identidad que permita una mejor gestión de la privacidad, evite comportamientos abusivos como el rastreo de usuarios, e integre mecanismos de confianza para validar las interacciones entre diferentes entidades. Idealmente, se busca reemplazar los sistemas actuales de gestión de identidad con una solución que preserve completamente la privacidad, permitiendo a los usuarios tener control sobre sus datos y herramientas para verificar la confianza en los diferentes servicios y proveedores de identidad. Además, se pretende reducir la influencia del proveedor de identidad, dejándolo solo como facilitador del material de autenticación, sin conocer para qué servicio está destinado ni siendo capaz de suplantar la identidad de los usuarios en caso de ser comprometido. Este es un objetivo ideal, ya que se reconoce la dificultad de establecer un objetivo tan ambicioso en el que actores poderosos como los proveedores de identidad perderían influencia y capacidad comercial en favor de los usuarios.

La metodología propuesta para alcanzar estos objetivos se basa en un enfoque sistemático y estructurado, dividido en seis objetivos específicos que se detallan a continuación:

O1 Analizar y estudiar las características y restricciones presentes en los sistemas actuales de gestión de identidad: El primer paso en la metodología es realizar un análisis exhaustivo de los sistemas de gestión de identidad existentes. Esto incluye la identificación de problemas, limitaciones y desafíos asociados con los sistemas tradicionales. Se realizará una revisión de literatura detallada y se evaluarán las características de sistemas como X.509 y SSO, así como las soluciones emergentes basadas en Blockchain. El objetivo es compilar una lista comprensiva de problemas a considerar en el diseño de nuevas soluciones.

O2 Analizar y estudiar los principales usos de las tecnologías DLT y sus mecanismos asociados: Se llevará a cabo una investigación detallada sobre las tecnologías DLT y su aplicación en sistemas de gestión de identidad. Esto incluye la evaluación de cómo las tecnologías como Blockchain pueden mejorar la confianza y la seguridad en la gestión de identidad. Se investigarán las diferentes implementaciones y se analizarán sus mecanismos para determinar su aplicabilidad y eficacia en el contexto de la gestión de identidad.

O3 Analizar las principales implementaciones actuales de las tecnologías DLT: Es fundamental estudiar las principales implementaciones actuales de las tecnologías DLT, como Shocard y Serto. Se evaluarán sus fortalezas, limitaciones y casos de uso para comprender cómo han abordado los desafíos de la gestión de identidad y qué áreas requieren mejoras adicionales. Este análisis permitirá identificar las lecciones aprendidas y las mejores prácticas para la creación de nuevas soluciones.

O4 Diseñar una solución para la gestión de identidad aplicando tecnologías distribuidas: Basado en las conclusiones obtenidas del análisis de los sistemas actuales y las tecnologías DLT, se diseñará una solución para la gestión de identidad que aplique tecnologías distribuidas. El diseño se enfocará en mantener al menos el mismo nivel de seguridad que las soluciones existentes, al tiempo que integra características avanzadas de privacidad. Se considerarán aspectos como la minimización de datos, la prevención del rastreo y la usabilidad para crear una solución integral.

O5 Diseñar una solución que combine el sistema de gestión de identidad distribuido con tecnologías DLT: En esta etapa, se diseñará una solución que combine el sistema de gestión de identidad distribuido con tecnologías DLT. El objetivo es mantener los niveles de seguridad y privacidad mientras se mejoran las características de confianza. La solución propuesta buscará integrar las mejores prácticas identificadas en el análisis de implementaciones actuales y diseñar un

sistema que aborde las deficiencias y aproveche las ventajas de las tecnologías descentralizadas.

O6 Verificar las soluciones de identidad obtenidas en escenarios reales: Finalmente, se realizará una verificación exhaustiva de las soluciones de identidad diseñadas en escenarios reales. Esto incluirá pruebas y evaluaciones para validar si las soluciones satisfacen las características deseadas en situaciones prácticas. Se evaluará la efectividad de las soluciones en términos de privacidad, confianza y seguridad, y se realizarán ajustes según sea necesario para mejorar la viabilidad y el rendimiento de las soluciones propuestas.

La metodología propuesta está orientada a proporcionar un enfoque integral y detallado para la mejora de la gestión de identidad digital. A través de un análisis exhaustivo, diseño de soluciones avanzadas y verificación en escenarios reales, se busca desarrollar un sistema que no solo cumpla con los requisitos actuales, sino que también establezca un nuevo estándar en términos de privacidad, seguridad y confianza en la gestión de identidades digitales.

CHAPTER 1

Introduction

This chapter is a brief introduction to digital identity, identity management (IdM) systems and distributed ledger technologies (DLT) in the context of privacy and trust relationships in current identity management solutions. It also outlines the gaps and shortcomings that have motivated this thesis. Next, it describes the main objectives and contributions. Finally, the chapter details the structure of this document and lists the publications that have resulted from the research carried out.

1.1. Contextualization

In an ultra-connected world through the Internet, data has become the real cornerstone. Smart cities, Industry 4.0, cloud applications, etc., are challenging traditional identity management systems, which are evolving more slowly by comparison. In addition, the emergence of algorithms that systematically analyse large volumes of data, the reduction of storage costs and the absence of good tools that allow users to control their privacy put users at risk. Location or even health data are collected for profit, often without the user being aware of the collection.

The concept of identity has different connotations, however, we can define it as the set of information known about a person. For example, a person's identity in the real world can be a set of attributes such as first name, surname, address, driving license, birth certificate, and others including elements such as the name, which is used as an identifier and allows us to refer to the identity without listing all elements. The driver's license or birth certificate, which are used as authenticators, are in addition issued by the competent authorities and allow us to determine the legitimacy of someone's claim

to identity. For example in the case of the driver's license, establishing permission to drive a motor vehicle.

The digital identity is the online equivalent of the real-world identity of a person or entity. It encompasses both the user's offline information, such as name, physical address, etc., and the image they project through their online activity.

Traditional identity management systems (IdMs) rely on the use of centralized identity providers (IdPs) that create, manage and maintain identity information of its users or smart devices and, at the same time, provide authentication mechanisms to service providers (SPs). This widely deployed solution enables the operation of single sign-on (SSO) technologies which are very convenient due to its simplicity. However, the exponential growth of online services has changed identity requirements and introduced new associated challenges. Impersonation or identity theft are security problems aggravated by characteristics such as mobility, temporality, or anonymity on the network. We are faced with the problem of determining whether our interlocutor is whom they say they are and whether their statements are true. Identity management becomes a critical element of guaranteeing security, privacy, and the correct functioning of services.

Traditional IdM systems generally offer basic privacy and user control functions. Users and services demand advanced security and privacy features while remaining user-friendly. In addition, new regulations (i.e., GDPR [3]) on the processing of personal data are imposing tough conditions on how personal data should be handled in order to protect users against organizations abusive behaviours such us the Facebook and Cambridge-Analytica scandal in 2010 [4].

Identity management systems are evolving towards decentralised models in order to improve their security features. In this sense, technologies such as distributed ledger technologies (DLT) [5] are taking the lead, Blockchain [6] being the most famous of these technologies thanks to the emergence of cryptocurrencies and more specifically Bitcoin [7]. In addition to cryptocurrencies, DLT technologies have proven to be valuable in other scenarios where digital identity and privacy preservation concepts are highly relevant such as border control, e-voting, e-residency, supply chains, etc. In any case, DLT scenarios still have to address numerous challenges [8] regarding linkability, network privacy, key management, or privacy regulations.

In addition to decentralised schemes, identity management systems are also moving towards self-sovereign models (SSI) [9, 10] where users take control over their data with the intention of reducing or eliminating the constant tracking, IdPs impersonations, or massive data leakages.

This thesis studies the evolution from traditional management systems towards a decentralised model in combination with the self-sovereign identity by combining privacy-preserving IdM systems with DLT or Blockchain technologies to provide a sufficiently robust and user-friendly solution that would maintain security standards while improving confidence in the entire infrastructure and reducing the chances of fraud.

1.2. Motivation and problem statement

With the rise of online services and the growth of connected elements (i.e, the IoT scenarios), the evolution of identity management systems is lagging behind. The need to protect users' privacy, secure communications and maintain trusted operating spaces has become a constant need to which there is little or no consistent response from traditional IdM systems. In contrast, new technologies such as those based on DLT (i.e. Blockchain) are gaining momentum and present themselves as an opportunity to improve not only business logic but can serve as a fundamental part of improving privacy, trust and security in digital identity management. This evolution means moving from centralised identity management models to distributed models where trust is no longer placed in a single central element but is spread across different nodes and even domains. Additionally, this evolution must be accompanied by compliance with the different data protection regulations that are being approved and applied today, such as the *General Data Protection Regulation* (GDPR) [3] in the case of Europe. Balancing privacy, trust, security and even business logic is proving to be a major challenge. Users increasingly demand to own their data, service providers need assurances to be able to function normally and today's identity managers are in an advantageous position where they can act almost freely and without control.

Traditional IdM (i.e., X.509, SSO) share a lack of solutions to increase user privacy often leaving data exposed, as in the case of X.509, or allowing an entity, in the case of SSO systems, to behave as a big brother tracking the actions of users or devices. Other solutions like Privacy-Enhancing Attribute-Based Credentials (P-ABCs) [2, 11, 12], are presented as a privacy-preserving solution making a similar proposition to X.509 but inheriting the management issues of X.509 and adding complexity. On the other hand, identity solutions based on DLT-Blockchain technologies like Shocard [13], Serto [14] and others tends to be incomplete by not providing a complete ecosystem. While traditional identity systems are weak on privacy, blockchain proposals lack sufficient tools (authentication, authorization) to complete identity management systems.

In this situation, an identity management solution must provide a sufficiently complete ecosystem to operate with at least the same traditional functions. It must also be aligned with data protection regulations and additionally provide the necessary tools for all parties involved to have their privacy, security and operational demands met. Therefore, the main requirements that a modern identity management system should encompass are:

R1 *Data-minimization.* Some identity management systems already have some basic data minimisation functions. However, it is essential to add advanced functions that allow personal data to be processed in a way that is adequate, relevant and limited in relation to the purposes for which they are processed.

R2 *Prevent users tracking.* Currently, identity management systems not only have no protection against user tracking, but often the identity providers themselves do

the tracking. In order to protect users' privacy, it is necessary to provide solutions that prevent identity providers from becoming a big brother.

R3 *Enhance trust among users, services and identity providers.* The trust in existing identity systems is mostly assumed by users to be inherent and simply works even if they are not certain that all the entities they interact with are who they expect them to be. An identity management system should have advanced trust features, which allow for a reliable verification of trust in the components of the existing infrastructure.

R4 *Minimise hardware requirements.* In some identity solutions, users are required to have specific hardware in order to be able to use credentials or other cryptographic material. This situation is not ideal and hampers the adoption of these solutions, so it is necessary to avoid this situation in order to achieve better adoption.

R5 *Integrable with existing standards and solutions.* In order for a novel solution to have an impact on the current scenario, it must take into account the compatibility of the most widely used technologies and standards such as OpenID, OAuth or SAML.

R6 *Transparency.* This requirement addresses how the solution affects users, identity providers and service providers. The solution must provide sufficient information about how personal data is protected and must be fully transparent in its operation.

R7 *Usability.* It is important to maintain usability levels of at least equal performance to the most widely adopted solutions. Even if security, privacy and trust features are better, if the user experience worsens, so does the adoption.

R8 *Compliance with personal data management regulations.* Policies for private data management are being tightened through regulations such as GDPR. Ensuring the alignment of the identity solution with these policies is a priority factor for the success.

1.3. Objective of this thesis

Although current state-of-the-art approaches provide valid solutions for identity management, they also present shortcomings and opportunities for improvement in terms of privacy, trust and security. These opportunities are especially related to requirements R1, R2, R3 and R8 which leave an area for further research and improvement.

In order to improve the performance of identity management systems and to provide a solution to the stated requirements, the objective of this thesis can be expressed as:

> *To analyse, design and validate solutions for digital identity management that include advanced privacy, trust and security features, maintaining or*

improving the security levels of existing solutions, without penalising usability and aligned with personal data protection regulations.

The priority objective is to offer a comprehensive identity management system that enables better privacy management, avoids abusive behaviour (such as user tracking), and integrates trust mechanisms to validate interactions between different entities. Ideally, the objective would be to replace current identity management systems with a fully privacy-preserving solution in which the users have control over their data as well as tools to verify trust in the different services or even identity providers. Furthermore, the influence of the identity provider would be reduced, remaining only as a facilitator of authentication material in which, it does not know which service it is intended for, nor is it capable of supplanting the identity of its users in the event of being compromised. In this case, we can speak of an ideal objective, because we are aware of the difficulty of establishing such an ambitious objective in which actors as powerful as identity providers would lose influence and business capacity in favour of users.

This thesis can be described through the following six specific objectives:

O1 Analyse and study the characteristics and restrictions present in current identity management systems in order to obtain a list of problems to be considered in the solution.

O2 Analyse and study the main uses of DLT technologies and their associated mechanisms to investigate their application in identity management systems in order to improve trust.

O3 Analyse the main current implementations of DLT technologies to learn about their strengths and limitations.

O4 Based on the conclusions obtained from **O1**, design a solution for identity management applying distributed technologies that maintains at least the same level of security as existing solutions and integrates advanced privacy features.

O5 Based on the conclusions obtained from **O2**, **O3** and with the results obtained obtained in **O4**, design a solution that combines the distributed identity management system with DLT technologies, maintaining security and privacy levels while enhancing trust features.

O6 Verify the obtained identity solutions, **O4** and **O5** respectively, to validate if the desired features are satisfied in real scenarios.

1.4. Contributions

In order to accomplish the objectives described in Section 1.3, this thesis provides the following contribution blocks.

- **To improve Identity Management capabilities evolving to a decentralized architecture.** This block of contributions defines an identity management solution that evolves the traditional scheme based on centralised identity providers towards a decentralised model. The main contribution of this block is the break with the traditional identity provider model and in particular the separation of the traditional identity provider (IdP) entitie among multiple partial identity providers, so that none of them alone can impersonate or track its users. In addition, it aims to facilitate integration with existing technologies such as OpenID or SAML and to offer user-friendly authentication based on familiar models such as the user-password method.

  Summarizing the main contributions:

  - Analyse the challenges associated with the use of privacy-preserving identity management solutions, addressed in the publication *OLYMPUS: towards Oblivious identitY Management for Private and User-friendly Services* [15].
  - Distributed identity management proposal. Main addressed requirements, the proposed architecture, an overview of the cryptographic building blocks and potential use cases. Contribution that can be found in the publication *The OLYMPUS Architecture* [16].
  - Analyse from a multidisciplinary approach some technical and legal foundations of proposal to build a privacy-preserving identity ecosystem. Publication *OLYMPUS: A distributed privacy-preserving identity management system* [17]
  - Analyse a standardisation proposal for p-ABC systems based on W3C Verifiable Credentials. Analysis in *Towards a standardized model for privacy-preserving Verifiable Credentials* [18]
  - A first implementation and performance review of the Pointcheval-Sanders Multi-Signature (PS-MS) scheme. Publication, *Implementation and evaluation of a privacy-preserving distributed ABC scheme based on multi-signatures* [19].

- **To improve the Trust Management by introducing DLT technologies into the IdM solution.** This block of contributions evolves the distributed identity solution with the objective of adding advanced trust management through DLT technologies. Firstly, it analyses the advantages and disadvantages of DLT technologies for privacy and studies the existing solutions for digital identity management. Finally, it makes an integrative proposal of the distributed IdM with DLT.

  The main contributions are:

  - Analyse the challenges associated with DLT technologies and privacy aspects. Publication *Privacy-Preserving Solutions for Blockchain: Review and Challenges* [8].

- Analyse the different existing solutions for digital identity management based on DLT systems. Publications *Privacy-Preserving Solutions for Blockchain: Review and Challenges* [8]. and *A Trusted Approach for Decentralised and Privacy-Preserving Identity Management* [20].
- To make an integrative proposal for distributed identity management with DLT technologies. Publication *A Trusted Approach for Decentralised and Privacy-Preserving Identity Management* [20].
- Evaluate the proposal and establish research opportunities for the future. Publication *A Trusted Approach for Decentralised and Privacy-Preserving Identity Management* [20].

1.5. Thesis structure

The this thesis is structured as follows:

Chapter 1 introduces the context, motivation, and objectives of this thesis. It outlines the key contributions of the research, describes the structure of the thesis, and lists related publications.

Chapter 2 provides a comprehensive background on identity management systems with enhanced privacy and distributed ledger technologies (DLT). It discusses innovative identity management projects and examines the role of identity management within the context of DLT.

Chapter 3 delves into privacy-preserving distributed identity management. It discusses the conceptual framework, objectives, requirements, processes, and architectural setup. This chapter concludes with a discussion on the primary goals, technological underpinnings, anticipated impacts, encountered challenges, and contemplated drawbacks of the proposed solution.

Chapter 4 explores a DLT-enabled identity management system designed to enhance trust. It provides an overview of the concept, objectives, and detailed processes and architecture of the system. The chapter concludes with a review of core objectives, technological advancements, potential impacts, challenges, and anticipated drawbacks.

Chapter 5 details the implementation and results of the proposed systems. It covers both non-DLT and DLT-enabled distributed identity provider implementations and examines specific use cases.

Chapter 6 concludes the thesis with a summary of findings and contributions. It also outlines future research directions to further enhance the DLT-based identity management framework.

1.6. Related publications

Within the framework of the work carried out in this thesis, publications in conferences, research journals and book chapters have been obtained. The most relevant

contributions are presented below, in chronological order.

Indexed Journals (JCR)

(P1) Bernabe, J. B., Canovas, J. L., Hernandez-Ramos, J. L., Moreno, R. T., & Skarmeta, A. (2019). **Privacy-preserving solutions for blockchain: Review and challenges.** IEEE Access, 7, 164908-164940 [8].

Blockchains offer a decentralized, immutable and verifiable ledger that can record transactions of digital assets, provoking a radical change in several innovative scenarios, such as smart cities, eHealth or eGovernment. However, blockchains are subject to different scalability, security and potential privacy issues, such as transaction linkability, crypto-keys management (e.g. recovery), on-chain data privacy, or compliance with privacy regulations (e.g. GDPR). To deal with these challenges, novel privacy-preserving solutions for blockchain based on crypto-privacy techniques are emerging to empower users with mechanisms to become anonymous and take control of their personal data during their digital transactions of any kind in the ledger, following a Self-Sovereign Identity (SSI) model. In this sense, this paper performs a systematic review of the current state of the art on privacy-preserving research solutions and mechanisms in blockchain, as well as the main associated privacy challenges in this promising and disrupting technology. The survey covers privacy techniques in public and permission-less blockchains, e.g. Bitcoin and Ethereum, as well as privacy-preserving research proposals and solutions in permissioned and private blockchains. Diverse blockchain scenarios are analyzed, encompassing, eGovernment, eHealth, cryptocurrencies, Smart cities, and Cooperative ITS.

(P2) Bernabe, J. B., David, M., Moreno, R. T., Cordero, J. P., Bahloul, S., & Skarmeta, A. (2020). **ARIES: Evaluation of a reliable and privacy-preserving European identity management framework.** Future Generation Computer Systems, 102, 409-425 [21].

Despite several efforts in the last years to make Identity Management Systems (IdMs) reliable, secured and privacy-respectful, identity-related cybercrimes are still continuously expanding. Current IdMs lack of proper security and privacy mechanisms that can holistically manage user's privacy, strong authentication and ID-proofing mechanisms based on biometrics, usage of breeder documents, while maintaining usability for mobile, online or face-to-face scenarios. To fill this gap, the ARIES EU project aims to set up a reliable identity ecosystem, combining mature technologies for meet highest level of assurance, such as biometrics or use of secure elements, with innovative credential derivation mechanisms. ARIES has devised and implemented a privacy-preserving and user-centric Identity Management framework as well as associated management practices that ensure usability and flexibility for identity management processes. This paper presents ARIES results obtained after the successful development and validation of the ARIES IdM System in the associated use cases.

(P3) Torres Moreno, R., Bernal Bernabe, J., Garcia Rodriguez, J., Kasper Frederiksen, T., Stausholm, M., Martínez, N., ... & Skarmeta, A. (2020). **The OLYMPUS architecture—Oblivious identity management for private user-friendly services.** Sensors, 20(3), 945 [16].

Privacy enhancing technologies (PETs) allow to achieve user's transactions unlinkability across different online Service Providers. However, current PETs fail to guarantee unlinkability against the Identity Provider (IdP), which becomes a single point of failure in terms of privacy and security, and therefore, might impersonate its users. To address this issue, OLYMPUS EU project establishes an interoperable framework of technologies for a distributed privacy-preserving identity management based on cryptographic techniques that can be applied both to online and offline scenarios. Namely, distributed cryptographic techniques based on threshold cryptography are used to split up the role of the Identity Provider (IdP) into several authorities so that a single entity is not able to impersonate or track its users. The architecture leverages PET technologies, such as distributed threshold-based signatures and privacy attribute-based credentials (p-ABC), so that the signed tokens and the ABC credentials are managed in a distributed way by several IdPs. This paper describes the Olympus architecture, including its associated requirements, the main building blocks and processes, as well as the associated use cases. In addition, the paper shows how the Olympus oblivious architecture can be used to achieve privacy-preserving M2M offline transactions between IoT devices.

(P4) Moreno, R. T., García-Rodríguez, J., Bernabé, J. B., & Skarmeta, A. (2021). **A Trusted Approach for Decentralised and Privacy-Preserving Identity Management.** IEEE Access, 9, 105788-105804 [20].

Identity Management (IdM) systems have traditionally relied on a centralized model prone to privacy, trust, and security problems, like potential massive data breaches or identity spoofing. Identity providers accumulate excessive power that might allow them to become a big brother, analyzing and storing as much data as possible. Users should be able to trust identity providers and manage their personal information straightforwardly without compromising their privacy. The European OLYMPUS project introduces a distributed approach for IdM based on enhanced Attribute-Based Credentials (ABC) that splits the role of Identity Provider to limit their influence and chances to become a unique point of failure. However, the trust relationship between service providers, users, and identity providers is still a gap in those kinds of privacy-preserving ABC systems. Decentralized technologies are an opportunity to break away from the centralized model and propose systems that respect privacy while increasing users' trust. This paper presents an evolution of the OLYMPUS architecture, maintaining all the privacy features and incorporating distributed ledger technologies to enhance trust and security in online transactions and IdM systems. The proposed system has been implemented, tested, and validated, showing its performance and

feasibility to manage user's identity in a fully privacy-preserving, distributed and reliable way.

(P5) Daoudagh, S., Marchetti, E., Savarino, V., Bernabe, J. B., García-Rodríguez, J., Moreno, R. T., ... & Skarmeta, A. F. (2021). D**ata Protection by Design in the Context of Smart Cities: A Consent and Access Control Proposal.** Sensors, 21(21), 7154 [22].

The growing availability of mobile devices has lead to an arising development of smart cities services that share a huge amount of (personal) information and data. Without accurate and verified management, they could become severe back-doors for security and privacy. In this paper, we propose a smart city infrastructure able to integrate a distributed privacy-preserving identity management solution based on attribute-based credentials (p-ABC), a user centric Consent Manager, and a GDPR based Access Control mechanism so as to guarantee the enforcement of the GDPR's provisions. Thus, the infrastructure supports the definition of specific purpose, collection of data, regulation of access to personal data, and users' consents, while ensuring selective and minimal disclosure of personal information as well as user's unlinkability across service and identity providers. The proposal has been implemented, integrated, and evaluated in a fully-fledged environment consisting of MiMurcia, the Smart City project for the city of Murcia, CaPe, an industrial consent management system, and GENERAL_D, an academic GDPR-based access control system, showing the feasibility.

(P6) García-Rodríguez, J., Moreno, R. T., Bernabe, J. B., & Skarmeta, A. (2021). **Implementation and evaluation of a privacy-preserving distributed ABC scheme based on multi-signatures.** Journal of Information Security and Applications, 62, 102971 [19].

Despite the latest efforts to foster the adoption of privacy-enhancing Attribute-Based Credential (p-ABC) systems in electronic services, those systems are not yet broadly adopted. The main reasons behind this are performance efficiency issues, lack of interoperability with standards, and the centralized architectural scheme that relies on a unique Identity Provider (IdP) for credential issuance. To cope with these limitations, this paper describes the first implementation of the Pointcheval-Sanders Multi-Signatures (PS-MS) crypto scheme proposed by Camenisch et al. and its integration in a distributed and privacy-preserving identity management system proposed in OLYMPUS H2020 European research project. Our efficient implementation provides remarkable privacy-preservation features for identity management in online transactions leveraging p-ABC systems, including unforgeability, minimal disclosure of personal data through zero-knowledge proofs, unlinkability in online transactions and fully distributed credential issuance across different IdPs, thereby removing the IdP as a unique point of failure. The performance of the implementation has been exhaustively

analyzed and evaluated with different curves, signers and number of attributes, and compared against Identity Mixer, the best-known p-ABC system, outperforming significantly the credential issuance and zero-knowledge proving and verification processes (2-4 times less execution time).

Conferences

(P7) Moreno, R. T., Bernabe, J. B., Skarmeta, A., Stausholm, M., Frederiksen, T. K., Martínez, N., ... & Lehmann, A. (2019, June). **OLYMPUS: Towards oblivious identity management for private and user-friendly services.** In 2019 Global IoT Summit (GIoTS) (pp. 1-6). IEEE [15].

The OLYMPUS EU project is addressing the challenges associated to the use of privacy-preserving identity management solutions by establishing an inter-operable European identity management framework, based on novel cryptographic approaches applied to currently deployed identity management technologies. In particular, OLYMPUS employs distributed cryptographic techniques to split up the role of the online IDP over multiple authorities, so that no single authority can impersonate or track its users. This paper describes the IdM ecosystem being developed in the scope of OLYMPUS, including its main building blocks, requirements and use cases.

(P8) Moreno, R. T., Rodríguez, J. G., López, C. T., Bernabe, J. B., & Skarmeta, A. (2020, June). **OLYMPUS: A distributed privacy-preserving identity management system.** In 2020 Global Internet of Things Summit (GIoTS) (pp. 1-6). IEEE [17].

Despite the latest initiatives and research efforts to increase user privacy in digital scenarios, identity-related cybercrimes such as identity theft, wrong identity or user transactions surveillance are growing. In particular, blanket surveillance that might be potentially accomplished by Identity Providers (IdPs) contradicts the data minimization principle laid out in GDPR. Hence, user movements across Service Providers (SPs) might be tracked by malicious IdPs that become a central dominant entity, as well as a single point of failure in terms of privacy and security, putting users at risk when compromised. To cope with this issue, the OLYMPUS H2020 EU project is devising a truly privacy-preserving, yet user-friendly, and distributed identity management system that addresses the data minimization challenge in both online and offline scenarios. Thus, OLYMPUS divides the role of the IdP among various authorities by relying on threshold cryptography, thereby preventing user impersonation and surveillance from malicious or nosy IdPs. This paper overviews the OLYMPUS framework, including requirements considered, the proposed architecture, a series of use cases as well as the privacy analysis from the legal point of view.

(P9) García-Rodríguez, J., Torres Moreno, R., Bernal Bernabé, J., & Skarmeta, A. (2021, August). **Towards a standardized model for privacy-preserving**

Verifiable Credentials. In The 16th International Conference on Availability, Reliability and Security (pp. 1-6) [18].

Lack of standardization and the subsequent difficulty of integration has been one of the main reasons for the scarce adoption of privacy-preserving Attribute-Based Credentials (p-ABC). Integration with the W3C's Verifiable Credentials (VC) specification would help by encouraging homogenization between different p-ABC schemes and bringing them all closer to other digital credentials. What is more, p-ABCs can help to solve privacy issues that have been identified in applications of VCs to use cases like vaccination passports. However, there has not been much work focusing on the collaboration between p-ABCs and VCs. We address this topic by establishing initial steps for extra standardization of elements that will help with the integration of p-ABCs into the standard. Namely, we propose a data model for predicates, which are a staple of p-ABC systems, and tools and guidelines to ease the adaptation process like a validation meta-schema. These ideas have been applied in a proof-of-concept implementation of the OLYMPUS distributed p-ABC scheme paired with serialization following the VC data model.

(P10) Bernabe, J. B., García-Rodríguez, J., Krenn, S., Liagkou, V., Skarmeta, A., & Torres, R. (2022). **Privacy-Preserving Identity Management and Applications to Academic Degree Verification.** In IFIP International Summer School on Privacy and Identity Management (pp. 33-46). Springer, Cham. [23]

This paper summarizes the contents and presentations held at a workshop at the IFIP Summer School on Privacy and Identity Management 2021, focusing on privacy-preserving identity management. In this document, we first introduce the necessary background on privacy-preserving identity management, including core cryptographic concepts. We then present a demonstrator scenario which benefits from the use of such technologies. Finally, we present a distributed privacy-preserving identity management framework offering an even higher level of security and privacy than previous work.

Book chapters

(P11) Bernabe, J. B., Torres, R., Martin, D., Crespo, A., Skarmeta, A., Fortune, D., ... & Alamillo, I. **An Overview on ARIES: Reliable European Identity Ecosystem**. Book: Challenges in Cybersecurity and Privacy - the European Research Landscape. Chapter: 11 Publisher: River Publishers. [24].

Identity-theft, fraud and other related cyber-crimes are continually evolving, causing important damages and problems for European citizens in both virtual and physical places. To meet this challenge, ARIES has devised and implemented a reliable identity management framework endowed with new processes, biometric features, services and security modules that strengthen the usage

of secure identity credentials, thereby ensuring a privacy-respecting identity management solution for both physical and online processes. The framework is intended to reduce levels of identity-related crimes by tackling emerging patterns in identity-fraud, from a legal, ethical, socioeconomic , technological and organization perspective. This chapter summarizes the main goals, approach taken, achievements and main research challenges in H2020 ARIES project.

(P12) Frederiksen, T. K., Hesse, J., Lehmann, A., & Torres Moreno, R. (2019, August). **Identity Management: State of the Art, Challenges and Perspectives.** In IFIP International Summer School on Privacy and Identity Management (pp. 45-62). Springer, Cham [25].

Passwords are still the primary means for achieving user authentication online. However, using a username-password combination at every service provider someone wants to connect to introduces several possibilities for vulnerabilities. A combination of password reuse and a compromise of an iffy provider can quickly lead to financial and identity theft. Further, the username-password paradigm also makes it hard to distribute authorized and up-to-date attributes about users; like residency or age. Being able to share such authorized information is becoming increasingly more relevant as more real-world services become connected online. A number of alternative approaches such as individual user certificates, Single Sign-On (SSO), and Privacy-Enhancing Attribute-Based Credentials (P-ABCs) exist. We will discuss these different strategies and highlight their individual benefits and shortcomings. In short, their strengths are highly complementary: P-ABC based solutions are strongly secure and privacy-friendly but cumbersome to use; whereas SSO provides a convenient and user-friendly solution, but requires a fully trusted identity provider, as it learns all users' online activities and could impersonate users towards other providers.

CHAPTER 2

Background and State of the Art

In this chapter, we dive deeply into the foundational technologies and concepts pivotal to the thesis's research. Our journey commences with an examination of cornerstone identity management technologies—namely, OpenID [26], OAuth [27, 28], SAML [29], PKI [30] and P-ABCs [12]. Each technology is dissected to reveal its integral role within existing systems, alongside an assessment of its inherent limitations. Transitioning from these traditional mechanisms, we pivot to the realm of Distributed Ledger Technologies (DLT). Here, DLT emerges as a transformative force, redefining the paradigms of secure and efficient identity management. We then explore how DLT-based identity management systems offer promising solutions to the limitations encountered in conventional methods. The chapter progresses to illuminate various pioneering projects, such as ARIES [21], ABC4TRUST [2], and PrimeLife [31]. These initiatives are scrutinized for their contributions towards innovating within the identity management landscape. Through this comprehensive exploration, we lay a robust groundwork that not only traces the evolution of identity management but also underscores its critical importance to the objectives of this thesis.

2.1. Identity management with enhanced privacy

In order to achieve the objectives outlined in **O1**, it is prudent to begin by evaluating existing identity systems. The concept of identity varies significantly across different fields, yet it fundamentally pertains to the question of "who a person is, or what characteristics differentiate him or her from others." Essentially, identity encompasses a collection of known information about an individual. Over time, however, this concept has broadened beyond merely human identifiers to include objects and devices.

This expansion is largely driven by the proliferation of connected devices—such as smartphones and smart TVs and cloud services, including streaming, storage, and online shopping platforms.

In the digital realm, both individuals and devices are frequently required to disclose private information to access internet services. For individuals, this might include personal details like address, age, and gender. For devices, this involves specific data such as manufacturer details, ownership, and operational parameters. The increasing interconnectivity and reliance on digital platforms underscore the urgent need for advanced digital identity management solutions. These systems must not only efficiently manage and protect identity data but also ensure it is used in a manner that respects privacy and enhances security.

Identity management (IdM) plays a key role in providing information about the user or device profile, service characteristics and access policies in order to improve the efficiency of other services and ensure the transparency of network operation, such as mobility and others. Not surprisingly, IdM systems are seen as an efficient way to provide trust between entities (users, network entities, services and devices), protect or mitigate the effects of malicious entities, manage user identities, identify entities in a system and control their access to different available resources. Nowadays, when we talk about digital identity management we are referring to multiple digital identities in which credentials such as passwords, OTP (One Time Password), PIN (Personal Identification Number) or digital certificates must be stored. As things stand now, most users have different user accounts (identities) linked to different services that require different sets of attributes. For example, Facebook wants pictures, age, location and in general every single data about the user. In the other hand, an online store wants to know the user full name, the address and usually if the user is over a certain age. These are just examples but most of people have many of such accounts and thus must repeatedly supply the same information or personal attributes to each of these providers.

There are two main issues to be addressed in identity management: (1) security and privacy, and (2) convenience and ease of use. Most problems with identity systems fall into one or both of these two categories. The IdM tries to address these issues by integrating solutions which includes the whole process of user identity creation, maintenance and deletion which is the life-cycle of identity. Every IdM system is composed of a core set of elements, namely: users, service provider (SP) and identity providers (IdP). The **User** is the client of both the SP and the IdP and can be a person, organisation, device etc. The **service provider** provides services to the user, relies on the identity material asserted by the IdP about the user and establishes access policies to its resources or services that may require certain attributes of the user (e.g. being of legal age). The **Identity Provider (IdP)** is the core concept of an IdM system. It provides identity and trust and has mainly two functions, firstly it must implement services for users such as user registration, identity verification and storage. Secondly, the IdP must process authentication requests from SPs and users. Figure 2.1 shows how the IdM components relates each other.

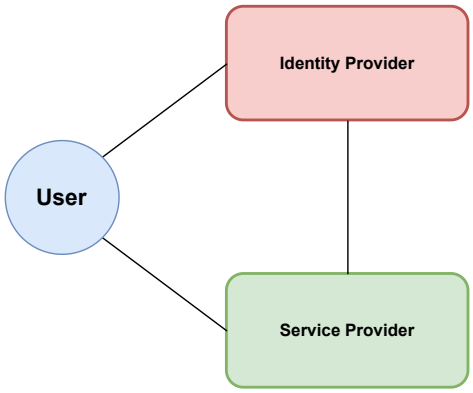


Figure 2.1: IdM Entities relation

A typical configuration for these entities is an Identity Provider providing authentication and attribute information to several Service Providers about a set of Users. These entities can be configured following different topologies or approaches, the most common are: centralised, distributed and hierarchical 2.2. In centralised topologies, a central entity concentrates the interactions with the other system participants. In distributed topologies, the workload is divided according responsibilities or other criteria. The high level of intercommunication of the distributed approach makes it highly resilient to outages or attacks aimed at compromising availability. Finally, hierarchical topologies follow an asymmetric distribution, with more elements at the edges, and fewer or only one, in the root. In this topology, the intermediate entities can alleviate the workload of the root node by answering requests too.

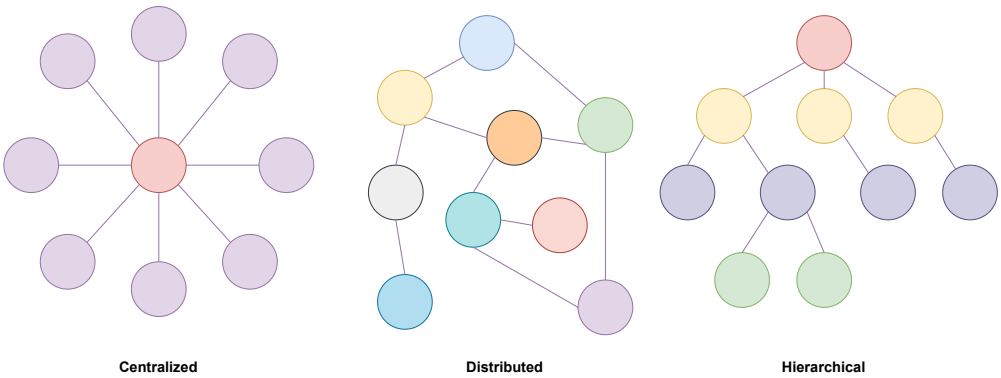


Figure 2.2: IdM Typical topologies

Identity management systems are typically implemented through centralized approach that replace or deeply integrate with existing login and access systems. They use a central directory of users, roles, and predefined permission levels to grant access rights to users or devices based on their roles and needs to access specific resources. The most important protocols or standards in the area of authentication and authorisation that are commonly integrated in identity management framework are presented below.

OpenID [26] is a unified user identification method published as an open standard that essentially acts as a single user identification system that can be used across multiple websites. It reduces the use of multiple user accounts on different service sites which often hampers the user experience, especially when trying to remember all the different combinations of usernames and passwords. OpenID allows users to log into virtually any website that supports the standard with a single ID, taking the agony out of the registration process and simplifying login to any affiliated website. In addition, it also acts as a personal data management system, so that when an end-user authenticates to a new service to register, they are prompted to indicate the data they wish to share with the new site.

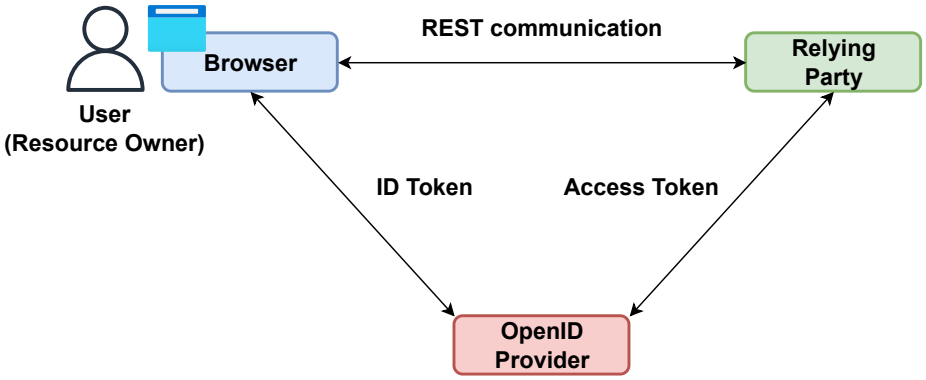


Figure 2.3: OpenID entities

OpenID defines three main entities (Figure 2.3): (1) The User Agent, which acts on behalf of the user (e.g., the browser) when the user wishes to make use of a service or resource. (2) The Relaying Party (RP), which is the service provider that relies on the OpenID provider to perform authentication and (3) the OpenID Provider, which is an OpenID authentication server that asserts that an end-user controls a certain identifier.

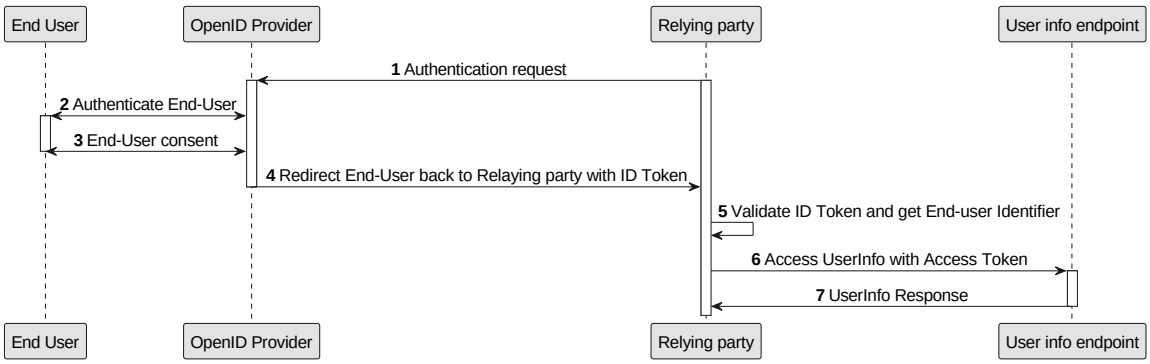


Figure 2.4: OpenID example flow

Figure 2.4 shows the typical operation flow in OpenID. In Step 1, the user attempts to start a session with your client app and is redirected to the OpenID Provider, passing

in the client ID, which is unique for that application. In Steps 2 and 3, the OpenID Provider authenticates and authorizes the user for a particular application instance. In step 4, a one-time-use code is passed back to the web server using a predefined Redirect URI. In Step 5, the client validates the ID token and the end-user ID. In Step 6, the web server uses the access token to get further details about the user (if necessary) and establishes a session for the user.

One particularity of OpenID is that it does not define which authentication mechanism should be used, so security depends on the trust placed in the OpenID Provider. If the OpenID Provider does not offer a good level of trust and the authentication mechanism is poor, it will not be recommended for services that require strong authentication. However, one of the main advantages of OpenID is the possibility for any user to set up their own authentication service, giving them greater trust and control over their personal data.

The main drawbacks of OpenID are essentially for security and privacy reasons. First of all, unifying all user identities carries significant risks as all user information is behind a single authentication process. In addition, the OpenID provider becomes a critical point through which all services must pass. This provider knows everything from the user's attributes to the sites and services he/she uses, and can act as a big brother. Moreover, in the worst case scenario, data leakage would compromise the entire digital life of its users.

Open Authorization, OAuth [27] is another protocol standard whose first version was approved approved in 2010, providing a method for clients to access server resources on behalf of a resource owner (such as a different client or an end-user). It also provides a process for end-users to authorize third-party access to their server resources without sharing their credentials (typically, a username and password pair), using user-agent redirections. The first version of OAuth was based on two existing proprietary protocols: Flickr's authorization API [32] and Google's AuthSub [33]. Over a few years of slowly adoption and integration, several specific areas were identified as needing improvement for reasons like they were limiting the framework or because the complexity of add new features.

The second major version, OAuth 2.0 [28], published in 2012, is a complete rewrite of OAuth 1.0 from scratch, sharing only the general goals and overall user experience. It does not maintain backwards compatibility with OAuth 1.0 or 1.1 and should be considered as a completely new protocol. It is supported by companies such as Google and Microsoft and is the current industry-standard protocol for authorization, focused on client developer simplicity while providing specific authorization flows for web applications, desktop applications, mobile phones, and living room devices.

The complexity of OAuth 1.0 signatures was a major problem for anyone coming from the simplicity of username/password authentication. The introduction of Bearer tokens in OAuth 2.0 provides a solution by simplifying the way APIs are interacted with by reducing the overhead of calls as only the token itself is needed to make requests. The figure 2.5 shows the main differences between OAuth v1.0 and OAuth v2.0 typical

flows.

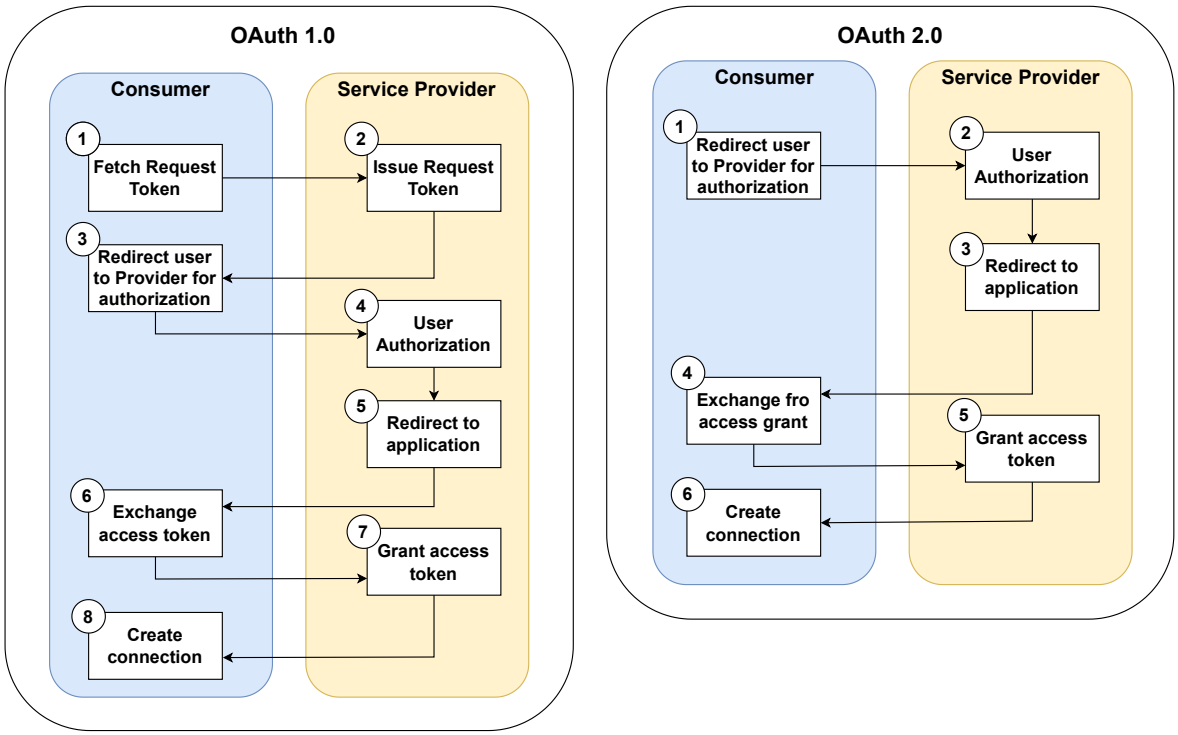


Figure 2.5: OAuth 1.0 vs OAuth 2.0

Bearer tokens are a simpler way of making requests because they do not require any cryptographic signing but, since the request contains a plain-text token that could be used by anyone if it is intercepted, all the exchanges must be made over a secure connection. The worst part is that there is nothing preventing other apps from using a Bearer token if it can get access to it.

Token management can be a problem. OAuth 1.0 used to issue access tokens of very long duration or even indefinite duration. Providers had to allow users to see which third-party applications were authorised to use their account and be able to revoke that permission if they wished. When revoking an application, access tokens issued for that application should no longer be accepted as soon as possible and depending on how this was implemented, this could be challenging or require additional links or steps. In contrast, OAuth 2.0, the authorisation server can issue a short-lived access token and a long-lived update token. This allows applications to obtain new access tokens without further user intervention, but also adds the ability for servers to revoke tokens more easily.

There are two main parts to OAuth 2.0: obtaining authorization by the user (the end result being the application has an access token for that user), and using the access token to make requests on behalf of the user. The methods for obtaining an access token are called flows. OAuth originally offered three distinct flows, web-based applications, desktop clients and mobile, which were eventually combined into a single flow that in

theory encompassed them. In practice, the experience was good only for web-based applications, and poor for the rest. OAuth 2.0 addresses this by defining multiple flows again, called *grant types*, with flexibility to support a wide range of application types. There is also a mechanism to develop extensions to handle use cases not previously thought of. In that sense, server-side apps use the "Authorization Code" grant type with a client secret, which prompts the user to authorize the application, and generates an authorization code that is handed back to the app. Single-page or mobile apps use the same grant type, but do not use the client secret. Instead, the security is in verifying the redirect URL. Moreover, OAuth 2.0 defines a "Password" grant type to allow applications to collect the user's name and password for exchange them for an access token but, it should not be used by third-party apps because they would have access to the username and password of the user.

Another key point of OAuth 2.0 is the improved scalability over version 1.0. Whereas OAuth 1.0 required managing state across different steps and often across different servers, requiring the generation of temporary credentials (often discarded without use), and issuing long-lived credentials that are less secure and more difficult to manage, OAuth 2.0 uses the client's credentials only when the application obtains the user's authorisation. After using the credentials in the authorisation step, only the resulting access token is used when making API calls, which means that API servers do not need to know the client's credentials, as they can validate the access tokens themselves.

Finally, OAuth 2.0 explicitly separates the roles of authorization server from resource server and defines four roles different roles instead of the three defined in OAuth 1.0 (figure 2.6): (1) client, (2) authorization server, (3) resource server and (4) resource owner, OAuth 1.0 uses a different set of terms where (1) client is known as the *consumer*, (4) resource owner is simply the *user*, and (3) resource server is the *service provider*. The separation of roles means that you can build out the authorization server as a standalone component which is only responsible for obtaining authorization from users and issuing tokens to clients. The two roles can be on physically separate servers, and even be on different domain names, allowing each part of the system to be scaled independently. The benefit to service providers is that the development of these systems can happen completely independently. They can be scaled, upgraded o even replaced without concerning the other parts of the systems.

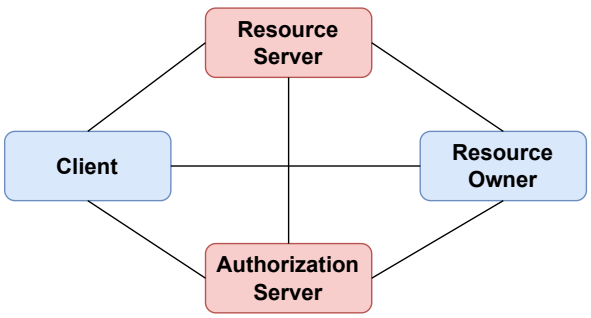


Figure 2.6: OAuth entities

Deciding on the most appropriate OAuth flow depends primarily on your type of application, but also on other parameters, such as the level of trust for the client or the experience you want your users to have. The typical scenario involves Authorization Code Flow 2.7 since normal web applications are service-side applications, which exchanges an authorisation code for a token.

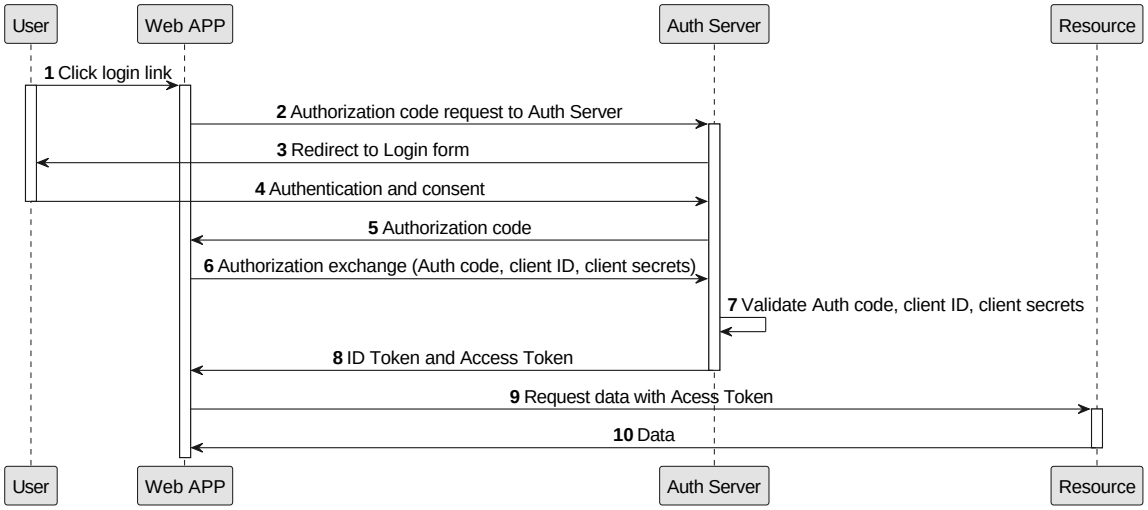


Figure 2.7: OAuth Authorization code flow

The user clicks Login (step 1) within the regular web application. The web-app redirects the user to the Authorization Server (step 2). After that, the Authorization Server redirects the user to the login and authorization prompt (step 3). The user authenticates using one of the configured login options and may see a consent page listing the permissions (step 4). Authorization Server redirects the user back to the application with an authorization code, which is good for one use (step 5). The web-app sends this code along with the client ID and secret to the authorization server (step 6) which validates them (step 7). The authorization Server responds with an ID Token and Access Token (step 8). Finally, the application can use the Access Token get information about the user (step 9) and the resource responds with the requested data (step 10).

Although OAuth was designed as an authorization protocol, its use has become widespread as an authentication mechanism as well. OAuth allows Clients to perform a pseudo-authentication of the Resource Owner based on the authentication performed by the authorization server prior to the authorization request. If the Client obtains the authorization grant, it means that the it has been successfully authenticated.

OpenID Connect (OIDC) [34] is an open authentication protocol that profiles and extends OAuth 2.0 to add an identity layer. OIDC allows clients to confirm an end user's identity using authentication by an authorization server. Implementing OIDC on top of OAuth 2.0 creates a single framework that promises to secure APIs, mobile native applications and browser applications in a single, cohesive architecture.

The main difference between OpenID and OAuth is that OpenID is an authentication protocol while OAuth is an authorization framework. OpenID and OAuth are both open standards that complement each other, but OpenID allows users to be authenticated by relying parties. An OIDC relying party is an OAuth 2.0 Client application that requires user authentication and claims from an OIDC provider.

OIDC enables scenarios where one login can be used across multiple applications, also known as single sign-on (SSO). For example, an application could support SSO with social networking services such as Facebook or Twitter so that users can choose to leverage a login they already have and are comfortable using. The OIDC flow looks the same as OAuth. The only differences are, in the initial request where a specific scope of openid is used, and in the final exchange where the Client receives both an Access Token and an ID Token.

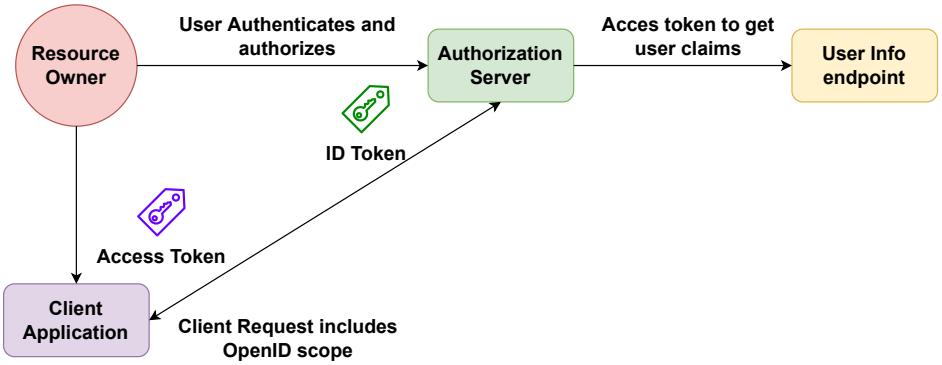


Figure 2.8: OIDC flow

OIDC integrates all the features of OpenID with OAuth 2.0 protocol. In this sense, OIDC incorporates in its specification the authorisation codes and access tokens defined in OAuth, adding a new third type of token, called *ID token*, which contains authentication information and can be used later to request additional information from the end-user. The specification incorporates the new endpoint called *UserInfo EndPoint*, which corresponds to a protected resource that returns claims (attributes) about the end-user information and where the client can request access by presenting the corresponding access token.

The OIDC flow (figure 2.8) starts with an OAuth flow that asks the user to authorize a request. As part of that flow, the client will include the OpenID Connect scope along with scopes for any additional information it wants about the user. After the request is processed, the client will receive an access token as well as an ID token issued by the authorization server that contains claims that carry information about the user. The user's SSO experience is made possible by the delivery of the ID token from the authorization server to the client. The client can then contact a special endpoint on the authorization server known as the UserInfo endpoint to receive the remaining claims about the user.

In summary, OIDC allows a user to authenticate with an external trusted identity

provider and augments the OAuth 2.0 framework towards an identity protocol by adding identity-centric concepts onto it to create a framework for distributed identity.

Security Assertion Markup Language (SAML) [29] is an open standard developed and approved by OASIS that allows identity providers (IdP) to pass authorization credentials to service providers (SP). SAML transactions use Extensible Markup Language (XML) for standardized communications between the identity provider and service providers.

SAML defines three main entities, shown in Figure 2.9. The User is the one who wants to access a resource or Web service, and therefore needs to be authenticated and, optionally, authorized. The Service Provider (SP) is responsible for offering the service to the User, and relies on the Identity Provider (IdP) to perform her identification and authentication. Besides the authentication process, the Identity Provider could provide service providers with additional information (role, age, etc.) about the User in the form of attribute statements.

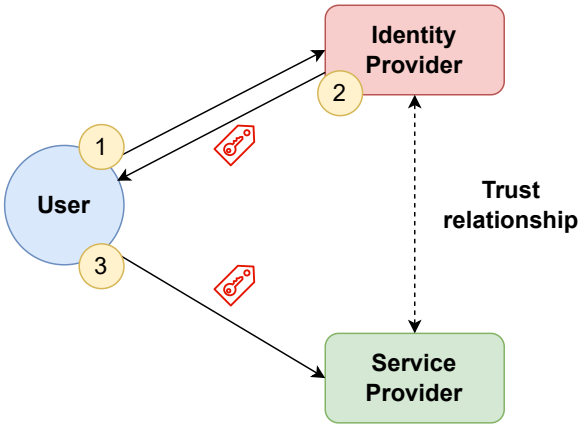


Figure 2.9: SAML entities

The SAML protocol establishes the link between authentication of a user's identity and authorisation to use a service. It simplifies federated authentication and authorisation processes for users, identity providers and service providers, allowing separate identity and service providers to exist, which centralises user management and provides access to SaaS solutions. In addition, it implements a secure method for passing user authentications and authorisations between the identity provider and service providers so that when a user connects to a SAML-enabled application, the service provider requests authorisation from the corresponding identity provider, and then the identity provider authenticates the user's credentials and finally returns the user's authorisation to the service provider so that the user can finally use the application or service. The protocol defines two main processes: (1) SAML authentication, which verifies the identity and credentials (i.e. username, password, etc.) of the user and (2) SAML authorisation to communicate to the service provider what level of access to grant to the

authenticated user. SAML also defines the authentication, attribute and authorisation statements. Authentication statements are issued by identity providers and inform the service provider about the successful authentication of the user. Attribute statements provide key value information related to the authenticated user, which can be used to make access control decisions, and finally, authorisation statements (deprecated since v2.0) assert that the user has been authorised to perform a given action on a specific resource.

SAML also defines several request-response protocols that allow service providers to request or query an assertion, request authentication of a subject, create and manage name identifier mappings to federate identities by linking accounts, and request a near-simultaneous logout of a collection of related sessions (single sign-on). SAML also provides definitions on how to transport SAML messages in a standard format through so-called SAML Bindings. In addition, it defines a set of SAML Profiles that address how the set of assertions, protocols and bindings can be used to solve specific use cases. Among them, the SSO web browser, figure 2.10 and enhanced client (ECP), figure 2.11, profiles plays an important role in SAML.

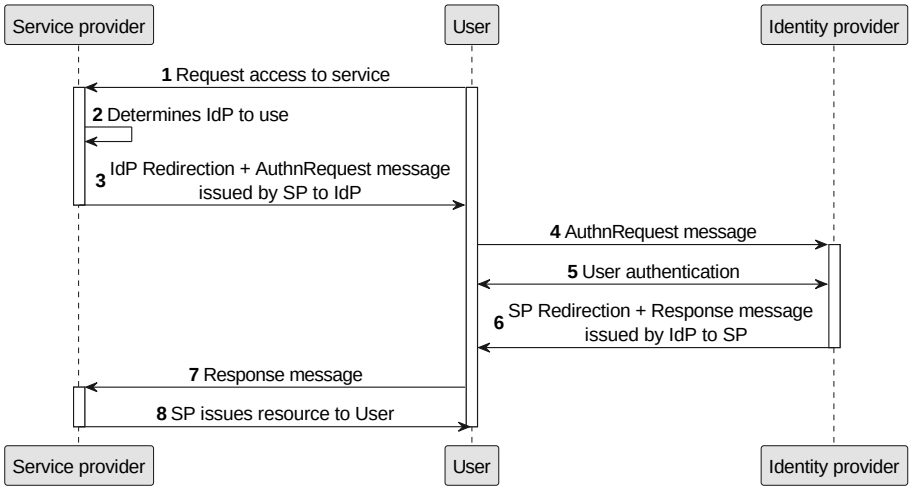


Figure 2.10: SSO web browser profile

When the User tries to access a secured resource at the Service Provider (SP), the User Agent (e.g., browser) sends a HTTP request to the SP asking for the specific resource. Once the request is received by the SP it determines, which Identity Provider (IDP) to be used for authentication. Once the IDP is selected the SP sends an Authentication Request to the IDP via the user agent with the use of either HTTP Redirect, HTTP POST or HTTP artifact binding. The User is identified by the IDP. If the same user has already logged once, the IDP may use an existing session. If the IDP receives an authentication request from a new user it establishes a new session. The IDP sends a response to the SP via the user agent, containing an error or Authentication Assertion if the user is valid. Finally, the Assertions of a valid user is sent to the Assertion Consumer URL of the SP and based on the Response from the IDP, the user

is granted the access to the resource or denied.

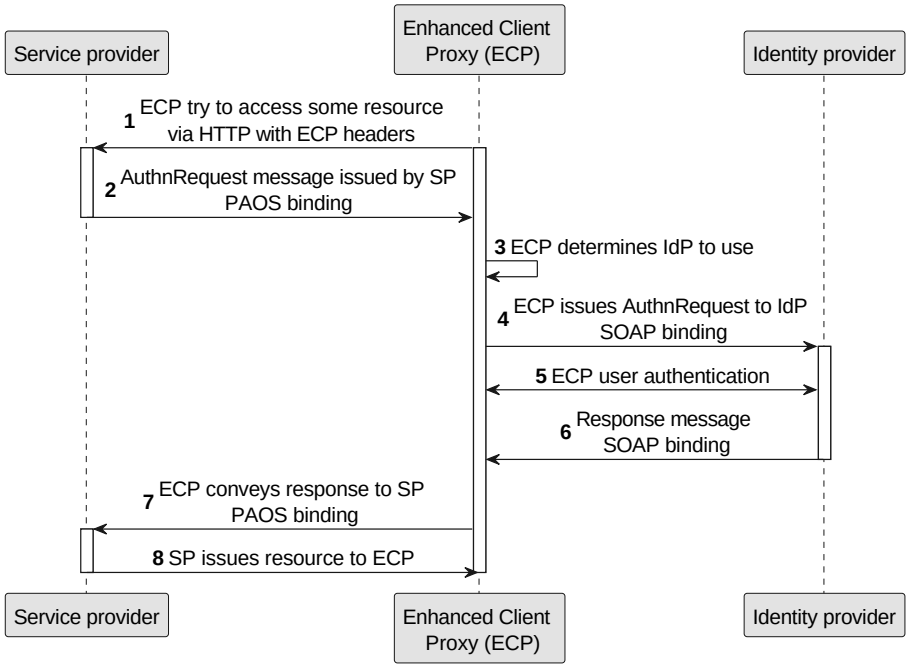


Figure 2.11: Enhanced client profile

When the ECP wants to access a secured resource at the SP, it sends a HTTP request to the SP mentioning that the Request is from an ECP. With the use of SAML Reverse Soap Binding (PAOS) the SP sends an authentication request to the Client. As the Client itself can select which IDP to be used, it selects an appropriate IDP to authenticate and sends an authentication request to the appropriate IDP using SAML SOAP binding. At this stage, the IDP identifies the client, and several messages may be exchanged between the IDP and the client. The details are not mentioned under the SAML ECP profile specification. We can see that this step is very similar to the Identification of a user in the SAML web based SSO profile. Once the IDP identifies the client, it sends an authentication response using SAML SOAP Bindings to the client targeting the actual receiver, the SP. The Client conveys the authentication response to the SP using PAOS bindings. Finally, based on the response, the SP grants the access to the resource.

Public Key Infrastructures (PKI) [30] are one of the best known and easiest to implement identity management methods. Its operation is based on the use of X.509 [35] certificates and the use of certificate authorities (CAs) that allow the issuance of certificates endorsed by trust chains through these CAs. One of the main advantages of this identity management scheme is that the CAs only have to be available at the time the certificate is obtained.

Any PKI consists of at least three elements (figure 2.12): Certificate Authority, the

relying party (RP) and the entity that must prove its identity (user, service, etc). The most typical use case is a web PKI where a CA has issued a certificate for a service (i.e., google.es) and a client (web browser) wants to verify the identity of the service or establish a secure connection.

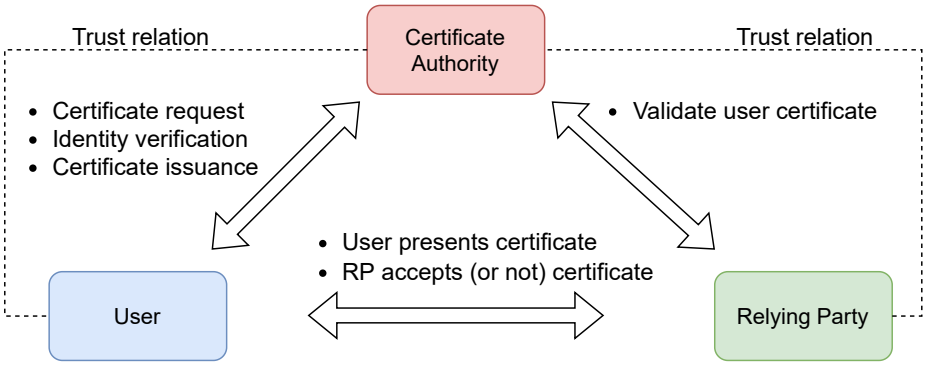


Figure 2.12: PKI Entities

PKI functions because of digital certificates, also known as a public key certificates. Those certificates use to cryptographically link ownership of a public key with the entity that owns it. They also enable the sharing of public keys to be used for encryption and authentication. The X.509 include the public key being certified, identifying information about the entity that owns the public key, metadata relating to the digital certificate and a digital signature of the public key the certificate issuer created. PKI governs encryption keys by issuing and managing digital certificates. Digital certificates have characteristics that make them very useful for authentication. They can act as the electronic equivalent of official documents (e.g. passport), they contain certified information about a person or entity in a tamper-resistant way. Can be traced back to the issuer, they have expiration date and can be revoked in case of necessity.

Certification Authorities (CAs) are responsible for the creation and issuance of digital certificates. They are also responsible for investigating or verifying the recipients before issuing certificates. These entities define the verification methods for certificate recipients, the types of certificates issued, their content and the security parameters or operations supported. CAs must formally document their issuance policies and once this is done, the certificate consumers are the ones who establish the degree of trust they have in the CA.

The process of creating a digital certificate is based on asymmetric cryptography. First, a public and private key pair is generated. Then, the CA requests the identification attributes of the private key owner and checks them. With the public key and attributes, a certificate signing request (CSR) is generated. The owner of the attributes signs the CSR request with his private key, thus proving his ownership. Finally, the CA validates the request and signs the certificate with the CA's own private key.

In this way, anyone can make use of the public part of the certificate to verify that it has actually been issued by the CA and confirming who is the owner of the private key

used to sign the certificate. Furthermore, assuming they consider that CA to be trusted, it is possible to verify that anything sent to the certificate holder will actually reach the intended recipient and that anything signed with the certificate holder's private key has actually been signed by that person or device.

In identity management scenarios, certificate authorities usually operate in a hierarchy of trust. This means that in addition to a root CA, there are other CAs to which the root CA has granted the ability to sign certificates on its behalf.

Despite the advantages, such as the possibility of having a digital clone of one's passport (or other official documents), the ability to digitally sign or even encrypt content by taking advantage of asymmetric cryptography, this system involves several risks for the security and privacy of users. The management of certificates by users is not user-friendly and usually needs the usage of specific hardware like smart-cards and smart-card readers which can lead to their loss or corruption, so they will not be able to log in their favourite services and will have to create a new account. Even worse, an attacker can obtain their certificates being able to completely impersonate the users. Finally, the X.509 is an all-or-nothing system and lacks the concept of minimal disclosure. This means that the user will always reveal the full content of his credential, including relevant and non-relevant data, during an authentication process.

Privacy-Enhancing Attributed-Based Credentials (P-ABC) [36, 37] appear as an answer to the privacy problems that suffer other approaches such as Public Key Infrastructure and the use of X.509 certificates. P-ABC systems follow the same approach as certificate-based PKI systems, i.e. users receive a credential containing a set of certified attributes (e.g. name, age, nationality...) that can be used to access a service provider or to convince another party of the validity of the attributes. The main difference compared to X.509 certificates is that it is no longer necessary to fully disclose the credential by exposing all attributes whether or not it is necessary. P-ABC credentials allow for a more fine-grained handling in terms of privacy, allowing the user to generate single-use tokens (presentation tokens) that expose only the minimum necessary information. The P-ABCs consist of three entities, figure 2.13, the user, the issuer and the verifier (usually integrated in a service provider). The issuer is responsible for issuing attribute-based credentials to a specific user. The User is responsible for securely storing the received credential and deriving presentation tokens from it. The presentation tokens generated from a credential are essentially mathematical proofs to assert that certain properties (predicates) are satisfied. Presentation tokens are sent to verifiers to check if they comply with a given policy and if they are cryptographically valid. For example, a user might have a credential that contains personal data (i.e., name and birth-date) along with medical data (i.e., blood type), with the P-ABC approach, the user is able to carry the credential in a personal wallet and generate presentation token which only reveals whether certain conditions are satisfied.

The most relevant features of P-ABC systems are the inclusion of advanced cryptographic technologies such as blind signatures (BS) [38, 39] and zero-knowledge proofs (ZKPs) [40], which represent a qualitative leap in the protection of users' privacy and are

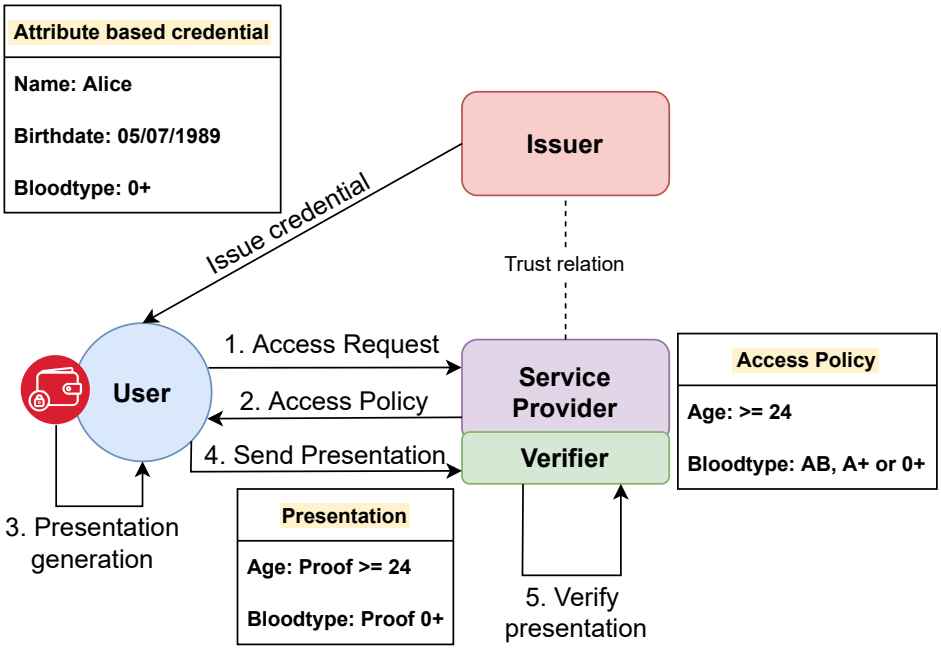


Figure 2.13: P-ABC scenario

the main advance and advantage over X.509-based PKI systems. The inclusion of these two technologies allows P-ABCs to obtain digital signatures over a set of attributes without the issuer having to know the content of what is being signed, thus adding a new layer of protection avoiding unnecessary disclosure of information.

The P-ABC systems set out seven key points in privacy protection.

1. Minimal disclosure. Presentation tokens do not reveal any of the attributes to be verified or those included in the credential.

2. Unlinkability. It is not possible to link a presentation token to the source credential and furthermore, it is not possible to link different presentation tokens to the same user.

3. Key binding. If a credential contains a key protected by a user secret, no presentation tokens can be created without the knowledge of that secret.

4. Advanced issuance. It is possible to issue new credentials based on the attributes of a previous credential without the issuer knowing the actual values of the attributes.

5. Pseudonyms. It is possible to create presentation tokens containing non-linkable pseudonyms.

6. Inspection. It is possible to encrypt values via presentation tokens that can later be revealed by a trusted party to, for example, demonstrate malicious behaviour.

7. Revocation. Credentials can be revoked so that no new presentation token can be successfully verified.

The advantages of P-ABC systems are their user-centric approach with advanced privacy preservation. Their offline approach allows the identity provider (issuer) to be online only during the issuance of credentials, providing usability in scenarios where connectivity is limited. In addition, they eliminate the need to contact the identity provider during authentication processes as in SSO systems, reducing the tracking that an IdP can do through user access requests. The user has full control over the information he/she discloses and the presentation tokens allow for minimal information disclosure. In terms of privacy, P-ABC systems are far superior to conventional SSO or certificates solutions. However, although P-ABCs are being available for almost 20 years with mature implementations such as Identity Mixer [41], U-Prove [42] or Persiano [43], the adoption was really scarce. The main reason has to do with the poor usability inherited from X.509 systems, which requires users to securely manage both their credentials and the associated cryptographic material. If an attacker gains access to this material, he/she could impersonate the user. The use of smart cards or other physical devices to secure these items is necessary to achieve an optimal level of protection but penalises adoption, users need to have adequate smart card readers and applications for all their devices. Moreover, the handling and verification of P-ABC tokens is more complex than that of conventional signatures and credentials. P-ABCs also requires specific advanced cryptographic building-blocks, such as zero-knowledge proofs, which are not available in regular cryptographic libraries. Instead, users and service providers must use specific software packages to analyse, create or verify such P-ABCs. Finally, the reliance on a single identity provider issuing attribute-based credentials should also be noted, which presents the issuer as a single point of failure, albeit less severe than in the case of SSO systems.

2.2. Distributed ledger technologies

The second objective **O2** of this thesis is to analyze and study the main uses of Distributed Ledger Technology (DLT) and its associated mechanisms to explore their potential applications in identity management systems, with a specific focus on enhancing trust. In this section, we will delve into distributed ledger technologies, examining their capabilities, benefits, and how they can be integrated into identity management frameworks to foster greater security and reliability.

Distributed ledger technologies or DLTs [5] are the fundamental basis for other technologies such as Blockchain. DLTs consist of a database that is independently maintained and updated by a number of members or nodes. These nodes constitute a network in which there is no central authority, i.e. there is no single node or entity that has the greatest responsibility. All participants must maintain a copy of the database they build independently. All nodes have access to a global registry (transaction log) where there is a history of the operations performed by all nodes, so that any node

can reconstruct the database and obtain its own results before adding a new entry (transaction).

A simple example would be four friends (the network), each with their own notebook (database), recording the outside temperature (data). Each time one of them makes a new entry, he/she communicates it to his friends through a group chat (transaction log) so that all of them can update their notebook later on. All participants must have access to the chat in order to keep their own notebooks correctly updated and none of them has more responsibilities than others. If someone new wants to join, it is only necessary to give him or her access to the group chat and he or she will be able to replicate the notebook independently.

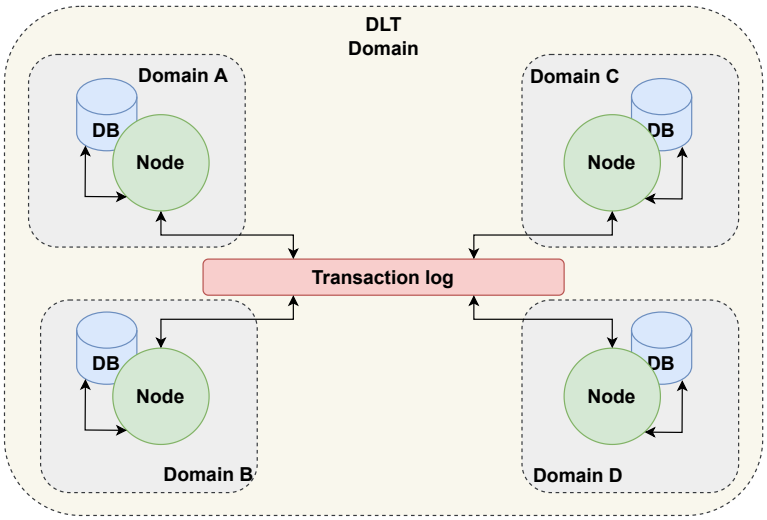


Figure 2.14: DLT nodes and domains example

DLTs are classified as public, private and permissioned, permissionless or any combination of the two.

1. **Public and Permissioned**. Allow anyone to deploy or use the DLT without the need to identify themselves or meet any requirements. However, the nodes forming the DLT network and running the deployed applications are authenticated and must be invited to join.

2. **Public and Permissionless**. It is the true decentralised system. No one has to notify, disclose their identity or meet any requirements in order to use DLT or be part of the network. Nodes can join and contribute freely and anonymously.

3. **Private and Permissioned**. In this model, there is no real decentralisation. Both applications and network nodes must have been invited to join the network, meet certain requirements or provide proof of identity. Any application or node can be removed at any time without notice.

4. **Private and Permission-less**. Applications must be invited to join the network and can be removed at any time without notice. Instead, nodes that make up the network and run applications can join and contribute freely and anonymously.

Within the distributed ledger technology there are different types, among which we can highlight Blockchain, Hashgraph, Directed acyclic Graph (DAG), Holochain and Tempo (RADIX).

Blockchain [6] appeared in 2008 as the fundamental basis for the first decentralised digital currency with cryptographic technologies, the Bitcoin [7] and it is the most popular DLT on the market where the transaction records are stored as a chain of blocks in a ledger (figure 2.15). Blocks are composed of a Block header and Block Data. The header contains a reference to the previous block in the chain (its predecessor), a timestamp, a nonce and the hash of the block data itself. In the block data we can find any kind of information such as a list of transactions or the public specification of a P-ABC credential. The timestamp makes the block impossible to be repeated in the future, as in addition to the time, the date of creation of the block is also stored, so there is no possibility of repeating the same hash. The nonce is a one-time random number used for authentication of data transfer between two or more parties. In Blokchain, the nonce works in combination with the hash as a control element to prevent manipulation of block information. This adds security and makes any change within the block, whatever it may be, impossible. This is because altering any element within a block alters the entire hash and its entire structure.

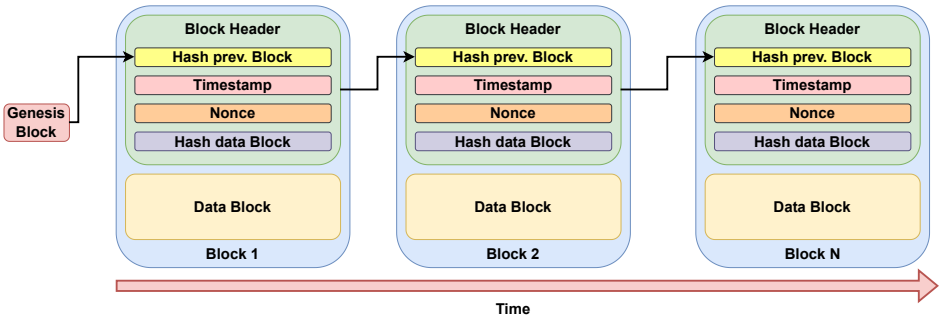


Figure 2.15: Blockchain block overview

In blockchain there are multiple blocks that get added to the ledger system. Based on the above example of temperature readings, the process can be summarised in five steps.

1. When one of your friends wants to make a transaction (Add a new temperature reading) he or she sends this information to all of his or her friends (nodes). Each one checks first some basic things, such as that this reading has not been sent before (i.e., duplicate transaction). It mostly depends on the nodes on that network. The nodes would have to come to an agreement that the transaction indeed took place.

2. If everything it is OK, each friend (node) saves the information about the transaction in a personal log (pool).

3. In established time periods, one of the friends is randomly chosen to propose a block containing the transactions in his/her personal log (pool). The consensus process start. The proposed block must be signed by one of the friends (node) and to see who signs, they must solve a mathematical problem whose winner will have the right to sign (i.e., in Bitcoin, this is achieved though a Proof of Work).

4. The winner of the previous process will sign the block, meaning that the block is sent to the rest of friends with a new version of the blockchain with all the blocks previously contained and a new block containing his/her own transactions. The block is given a unique ID before it is placed on the ledger.

5. Finally the rest of the friends (nodes) will update their copies and the chain will be completely updated with the fresh block.

An important aspect of the functioning of the Blockchain (and DLT systems in general) are the consensus algorithms [44,45]. The consensus is a problem in distributed computing where nodes must reach an agreement to add new transaction records to the ledger. In Blockchain the structure is designed to be valid in a trustless and unreliable network with adversarial users. The best known consensus protocols are Proof of Work (PoW), Proof of Stake (PoS), Proof of Importance (PoI) and Virtual Voting. However, depending on the solution adopted and the deployment scenario, it is possible to find other types of consensus algorithms derived from these or even completely new ones (i.e., Proof of Weight, Proof of Capacity, Proof of Burn etc).

- Proof of Work (PoW). It is the best known type of proof. It was introduced in Bitcoin [7] and is still used today. In this method, the computer does computations to solve a mathematics puzzle related with the Hash function. Hash is used for confirmation of the transactions stored in blocks. A miner, that is the computer trying to solve the hash, will try to find a specific value as a nonce in such a way that the hash value meets a predefined condition. In PoW, to reach consensus in the network, miners try to find hash value equal to or smaller than a certain given value. The advantage of this system is the high security, decentralization and acceptable levels of scalability but, in the other hand, the function of mining and validating blocks wastes huge amount of energy, the throughput is a problem in fast-growing scenarios, the block creation time is high and it is hardware dependant due to the computational cost.

- Proof of Stake (PoS). Proof of Stake is one of the most widely used consensus algorithms. The PoS algorithm is based on the creator's choice of the next block through random selection combinations. The node selected to make the next block will be chosen through a quasi-random process in which the selection depends on the assets stored in the wallet related to that node. PoS does not

need high computational power to validate any proof and therefore miners will not receive any reward (only transaction fees). However, although not as much power is needed as in PoW, there is a dependency on the nodes that have the most stake, centralising the process in some way. In addition, there is another problem called "nothing at stake", which means that if a node has nothing in its stake while misbehaving, it has no fear of losing anything. Therefore, there will be no obstacles for the node not to misbehave. The advantages of PoS are fast block creation, higher throughput and better energy efficiency. In the other hand, PoS suffers problems related with the centralization that can occur and the lower cost of misbehaving.

- Proof of Importance (PoI). Proof of Importance is the mechanism firstly introduced by NEM [46], that is used to determine which network participants (nodes) are eligible to add a block to the blockchain, a process that is known by NEM as "harvesting". In exchange for harvesting a block, nodes are able to collect the transaction fees within that block. Accounts with a higher importance score will have a higher probability of being chosen to harvest a block. In order to be even eligible for the importance calculation, the NEM protocol requires that an account hold at least 10,000 vested XEM. Three factors are taken into account to calculate the importance of a node: (1) Acquisition, (2) Transaction association and (3) Number and size of transactions in the last 30 days. After the score is calculated based on the previous factors, it will receive an opportunity related to the score achieved to add a block. Unlike PoS, this method guarantees decentralisation. The PoI model is fast and energy efficient. There is no need for a mining process as in PoW and the scoring election system ensures decentralisation. Furthermore, it does not require specific or particularly powerful hardware, which makes this system a breakthrough in consensus mechanisms.

- Virtual Voting. This consensus algorithm is introduced in the DLT Hashgraph [47] solution. All members have a copy of the hashgraph so that a node can calculate what vote a neighbouring node would have sent it. Unlike a traditional Byzantine protocol [48], here it is not necessary to send any vote because the nodes. All members can reach a Byzantine agreement on any number of decisions, without a single vote being sent. This solution has significant advantages in terms of bandwidth and decision speed.

Continuing with DLT types, **Hashgrapgh** [47] is another popular one. It is a permissioned solution organised as a structure that has columns and each member is represented by a column in the network. All columns have many vertices. Each of the vertices is called an event (ledger record) . Users of the network perform two types of actions: (1) The user can create an event and send it, (2) The user randomly chooses a member of the network and gossips all the information he/she knows, i.e. sends the information to the member about the event creation. The distribution of events takes place with the help of the gossip-about-gossip protocol, in a nutshell, once a transaction

occurs, neighbouring nodes share that information with other nodes, and after a while all nodes would learn about the transaction. This process is quite fast, so it would only take a few minutes for all members of the network to learn about the event. In Hashgraph, events or ledger records store four different types of information:

1. Hash of another user event.
2. Hash of the previous event of the user.
3. Zero or more events sent by the user.
4. Timestamp at which the event was created.

In the gossip-about-gossip protocol the user digitally signs the event and gossips about it. The two hashes included in the event allow members to know where the event originated from and where it was directed to. Hashgraph uses this information to build a directed acyclic graph (DAG) of events, figure 2.16,that is updated as events are gossiped on the network. Finally, the signature of the event helps to identify the creator and prevent tampering.

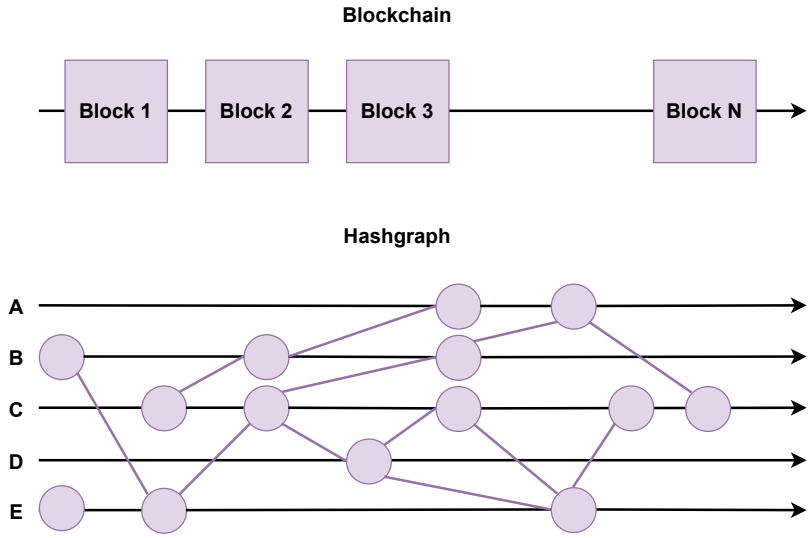


Figure 2.16: Hashgraph vs Blockchain

In Blockchain, a miner is able to choose which transaction to include. For example, if both you and one of your friends have taken temperature readings and you are waiting for your transactions to be verified, other nodes could selectively choose your friend's transaction first even though yours has been done before. Meanwhile in Hashgraph the verifier nodes have to include both your transaction and your friend's transaction in the way they have been carried out to avoid anyone being left behind. This solution benefits directly from connection speed as the faster the connection, the more transactions and the faster you can operate.

Another important feature of Hashgraph is its Byzantine fault tolerance (BFT) [48]. In a BFT system, no group or entity can influence the achievement of consensus and furthermore, once consensus is reached, all members will know that it has been reached and it will remain unchanged. In a DLT system, all nodes have a database that shares similar properties, yet the nodes are never sure that consensus has occurred. In Hashgraph, nodes can be sure that consensus has occurred, bringing atomicity, consistency, isolation and durability (ACID) to this solution. The way consensus is achieved is through the virtual voting protocol [47], in which all members maintain a full DAG of events. Since it is easy to know how a node will vote, as each node has all the information of what each node knows and when it knows it, this is used to find the order of transactions. Building a DAG on each node which helps to achieve consensus on the correct order independently and saving bandwidth, as nodes do not have to transfer their vote to other nodes. This is because other nodes have the necessary amount of information on how a node will vote in the election.

Direct acyclic graph (DAG) [49] is an ambitious proposal that emerges as an alternative to Blockchain. Hashgraph [47] and IOTA [50, 51] are the most well known solutions based on it. This solution provides the benefits of Blockchain and even enhances them through a completely different architecture, figure 2.17. The DAGs are made up of vertices and edges. The direction of the lines heads in one direction. They are acyclic, which means that the vertices do not loop back on themselves and it means that if you start at one point, you cannot return to the same point. In DAG, vertices

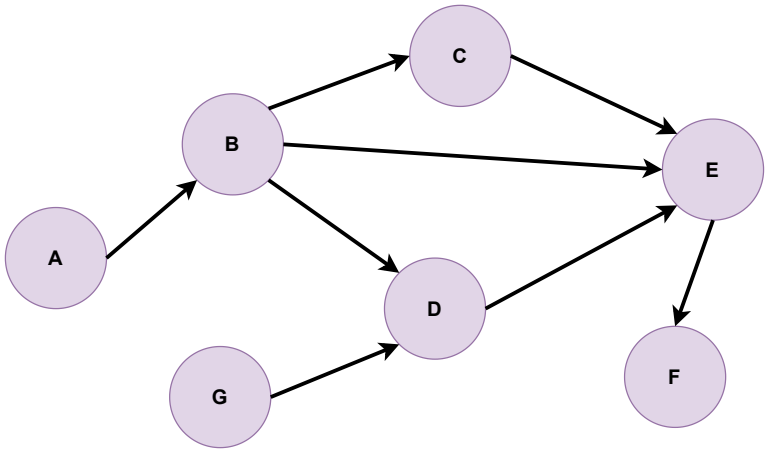


Figure 2.17: Direct acyclic graph

represent a transaction. It does not use blocks to store information but makes use of a node or a group of nodes that are developed simultaneously. It performs a small proof-of-work operation when a node sends a transaction and when a transaction is to be added, it builds on the older ones. The distributed ledger system stores transaction processes on the nodes. All nodes in the network validate transactions that are in turn represented by other validated transactions. Any node can initiate transactions, but to validate them, it must first validate at least two previous transactions, so the

more transactions a node validates, the more will be validated for it. Once validated, transactions are committed. Moreover, the more transactions a branch has, the more weight it will have on the DAG and finally, to prevent nodes from validating only their own transactions while ignoring the rest, an algorithm randomly selects which transactions to validate.

Compared to Blockchain, DAG has notable differences in structure, mining, validation and transaction speed. DAG is structured as an acyclic graph where each transaction is independent. In this structure, previous transactions validate the next one so that consensus is reached. In addition, transaction validation requires miners to have validated at least two previous randomly assigned transactions, preventing miners from postponing or even cancelling a transaction as could happen in blockchain. Regarding scalability, in blokchain this is a serious problem, but nevertheless in DAG, due to its structure, the larger it is, the more transactions per second it can perform.

Holochain [52] is presented as the evolution of Blockchain. It is a DLT that is distributed among nodes to avoid any instance of centralised control of the data flow. Whereas Blockchain aims to decentralize network transactions, Holochain aims to decentralize the interactions between individual nodes as well. Each node on the network runs its chain, allowing them to operate independently while still being a part of a larger network that includes thousands of other similar nodes. Figure 2.18 shows the difference between Holochain and Blockchain architecture.

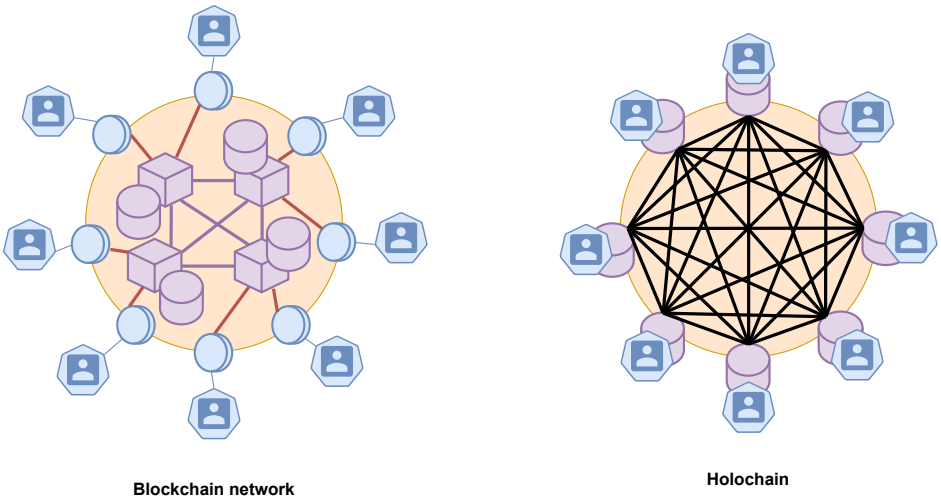


Figure 2.18: Holochain vs Blockchain

In Holochain users are the core concept. All the system is modeled from a user perspective in an agent-centric computing. Each user runs their own copy of the backend code, controls their identity, and stores their own private and public data. An encrypted peer-to-peer network for each app means that users can find each other and communicate. This way of operating implies that all participants know the rules necessary to operate, as they have their own copy of them. This makes it possible to check the data of other participants and verify that they comply with these rules.

Finally, Holochain also provides cryptography to prove authorship and prevent data manipulation. Holochain provides intrinsic data validity.

To prevent data from being lost or falsified at the time of creation, Holochain includes peer witnessing process. This means that each time public data is published, it is validated and stored by a random selection of nodes (witnesses). Together, participants are able to detect modified or invalid data. They spread evidence about malicious peers and take measures to counter threats.

In Holochain, each node operates its own chain independently, which means that miners are free to operate autonomously on what Holochain calls a distributed hash table (DHT). In this table users can store data, which is actually distributed in different locations, using certain cryptographic keys.

RADIX [53] is a solution that does not make use of Blockchain although it preserves the sequence of the information in the log, in addition to the timestamp (Figure 2.19). This type of DLT is based on three pillars: (1) having a group of networked nodes, (2) having a global registry distributed among the group of nodes and (3) special algorithms for timestamping the events.

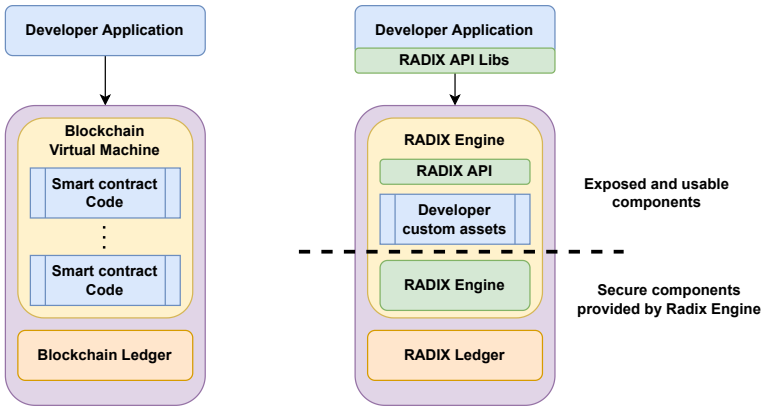


Figure 2.19: RADIX vs Blockchain

The most fundamental unit of the Radix stack is the Radix Node. The Node implements a 3-phase byzantine fault tolerant consensus protocol allowing an open network of nodes to quickly and safely decide on and commit correct transactions to a distributed immutable ledger. The Node also includes a layer called the Radix Engine that defines the application functionality of the network by creating and validating transactions. Radix has practically infinite linear scalability; while preserving cross-shard atomic composability. To do so, introduces a consesus protocol called Cerberus [54, 55]. Cerberus achieves this by starting with a unique data structure which is partitioned in a process called sharding. Each partition is called shard and independently operate as blockchains. Cerberus apply a sharding process to split up the ledger into 2256 shards. This is large enough to fit every thousandth atom in the observable universe into its own shard; or for another comparison, all possible combinations of Bitcoin addresses could fit in the Cerberus "shardspace" 79 billion billion billion times. Every change

to the ledger, or "substate", is deterministically allocated its own shard based upon its hash. When a transaction occurs, Cerberus allows nodes to temporarily "braid" consensus across the shards of relevant substates together. Related substates can thus be composed into atomic transactions when needed, and unrelated substates can be processed completely in parallel. Each node is only required to serve a subset of shards - no global state or ledger is maintained by any one node.

Because of this, as nodes are added to the network, transaction throughput increases linearly without practical limit; transactions reach settlement finality in less than five seconds; transaction fees will always be tiny; nodes can always be run on simple hardware; and the ability to compose transactions atomically across the global ledger will never be sacrificed.

2.2.1. Identity management in distributed ledger technologies

Objective **O3** outlines the implementations of DLT technologies, strengths and limitations. While distributed technologies are emerging strongly, and applications such as Blockchain are taking identity systems to a new level where privacy and security are the challenges to address [8, 56]. Proposals in the context of the Blockchain are growing in number, driven by the rise of cryptocurrencies. Hawk [57], Zcash [58] or Zerocoin [59] are cryptocurrencies that already add privacy features such as zero-knowledge proofs or linkability control. Privacy-preserving solutions based on crypto-privacy techniques are emerging to empower users with mechanisms to become anonymous and take control of their data following a Self-Sovereign Identity (SSI) model. In that sense, solutions such as Sovrin [60], Serto (previously uPort) [14], Shocard [13] and Hyperledger Aries [61] are some of the foremost proposals.

Sovrin [60] is an identity management solution that runs on top of permissioned blockchain, in particular, Hyperledger Indy [62]. Sovrin supports DPKI (Decentralized Public Key Infrastructure), where every public key has its public address in the ledger (DID, decentralized identifier [63]) that enable universal verification of claims. Users can have different DIDs for each existing relationship, with different key pairs. Sovrin allows attestation, verifiable assertions, and anonymous credentials based on zero-knowledge proofs, with the scheme proposed by Camenisch-Lysyanskaya [64]. The Sovrin approach is very comprehensive, and its advantages, such as unlinkability, identity recovery, integration of DIDs, or zero-knowledge proofs, are well integrated. However, Sovrin does not provide an authentication service and lacks usability by not displaying clear and precise information on the privacy implications that may arise. Moreover, it does not support smart contracts, which is an explicit limitation of the scenario. As for the credentials used, the underlying cryptography is old, negatively impacting its efficiency.

The Sovrin architecture can be summarized as figure 2.20 shows. The core element is the Sovrin ledger entity which contains transactions associated with specific identifiers. The ledger is written, distributed and replicated through the stewards nodes. Each of

them runs a specific byzantine fault tolerant protocol called Plenum [65]. The usage of a permissioned ledger implies that there is no need to use expensive proof-of-work computations to reach consensus. On the other hand, trust in Sovrin starts from the common root-of-trust formed by the distributed ledger, but as new organisations and users join the network, they can become trust anchors (i.e. allowed to add more users and organisations).

Users interact with Sovrin through a mobile application and control software agents acting on their behalf. The agents are network endpoints that are always addressable and accessible. Agents also provide a backup service and encrypted storage of attribute credentials. The mobile application also helps users manage cryptographic keys, which are stored on the users' mobile device. Finally, as recovery system, Sovrin offers a mechanism that relies on the user selecting a set of trustees. When requested to do so by the user, a specified quorum of trustees must sign a new identity record transaction that stewards must verify.

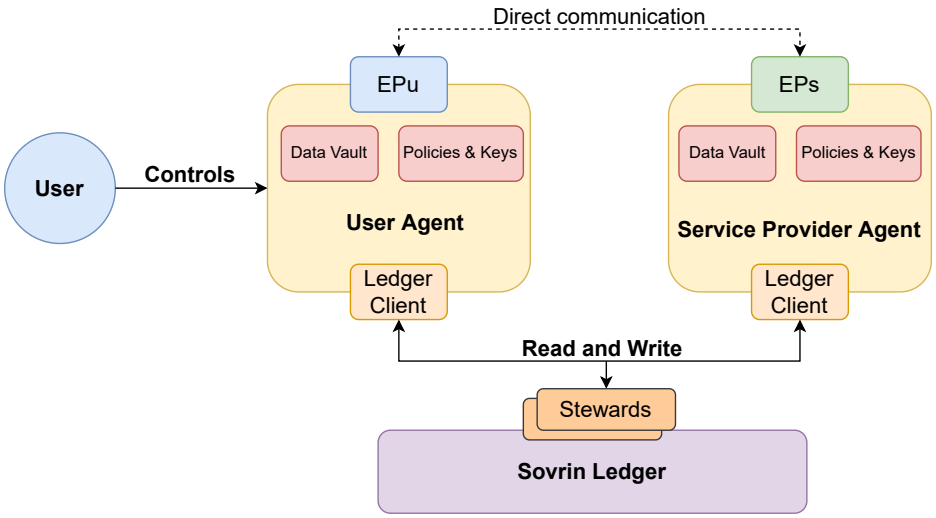


Figure 2.20: Sovrin overview

In Sovrin, each user can select from the attribute credentials what they wish to share with a relying party. This is made possible through the use of anonymous credentials. Although users can choose to store those attributes on the ledger, in general, they will prefer to use the storage capabilities of their mobile phone or their agent to transmit attributes to other parties through secure communication channels and use the ledger to identify the correct network endpoint to use. The use of attribute-based credentials allows users to only reveal credentials that they choose. However, verifying the relying party with whom data is shared is an unsolved problem. Sovrin is trying to alleviate this problem through a web of trust managed by the Sovrin Foundation. Nevertheless, in Sovrin, users must trust the agencies that will act on their behalf in the Sovrin network and the administrators that maintain the ledger. Depending on the choice of the agent and its implementation, a lot of information could potentially be in the hands

of third parties. Finally, another important aspect that remains unresolved in Sovrin is the user experience, which is still not simple enough to make the adoption of the system a success despite its security qualities.

uPort [14] is another identity solution that works on permissioned blockchains. It depends on Ethereum [66, 67], so the essence of the uPort identity is the Ethereum account address on which users interact, and the identity is permanent. It uses a 20-byte hexadecimal identifier to represent the user's ID, with the address of a Proxy Smart contract deployed over the Ethereum network. The smart contract is used as an indirection method between the user's private key (hosted on their device) and the accessed service. The user's application contacts a smart contract that contains the access control logic. This system provides some unlinkability by the possibility of having different user IDs. In addition, it adds selective disclosure with the possibility of attribute encryption. Finally, it additionally supports identity recovery in loss and integrates with the decentralized identifier (DID).

Figure 2.21 shows the uPort architecture. Two smart contracts composes the uPort identity: controller and proxy. To create a new identity, the user through the uPort app creates a new asymmetric key pair and sends a transaction to the Ethereum network that instantiates a controller that contains a reference to the newly created public key. Then, a new proxy is created containing a reference to the new controller address. The controller is the only one capable of invoke functions of the proxy contract. The address of the proxy comprises the unique uPort identifier (uPortID) of a user. A user is free to create multiple uPortIDs that are unlinkable. The registry is another smart contract that provides a decentralised mapping of uPort identifiers to identity attributes, which can be globally accessed for reading. In addition, the registry can refer to off-chain storage such as IPFS [68].

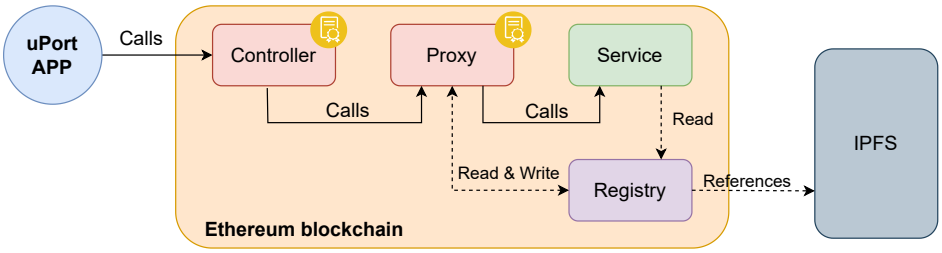


Figure 2.21: uPort overview

uPort puts more control over uPortIDs in the hands of users, but at the same time increases the responsibility of users and adds an extra layer of complexity for users. uPort does not require disclosure of personal data to create a uPortID for restricted use and also respects privacy in terms of the inherent lack of linkage between uPortIDs. However, the registry (if used) represents a centralisation point that can be probed for information on identifiers and identity data. There is a possibility that over-reliance on the registry could compromise privacy. In addition, uPort does not perform any

identity proofing. uPort simply specifies the format of attributes that are stored in its registry. As consequence, an uPortID owner having write-access to their own respective part of the registry can selectively discard negative attributes.

Shocard [13] is an identity management solution that leverages DLT to bind a user identifier, an existing trusted credential (e.g., passport, driver's license), and additional identity attributes, together via cryptographic hashes stored in Bitcoin transactions, figure 2.22. Shocard uses a central server as part of its issuing scheme, which mediates the exchange of identity information between the user and the relying party. Shocard proposes a three-phase scheme: bootstrapping, certification and validation.

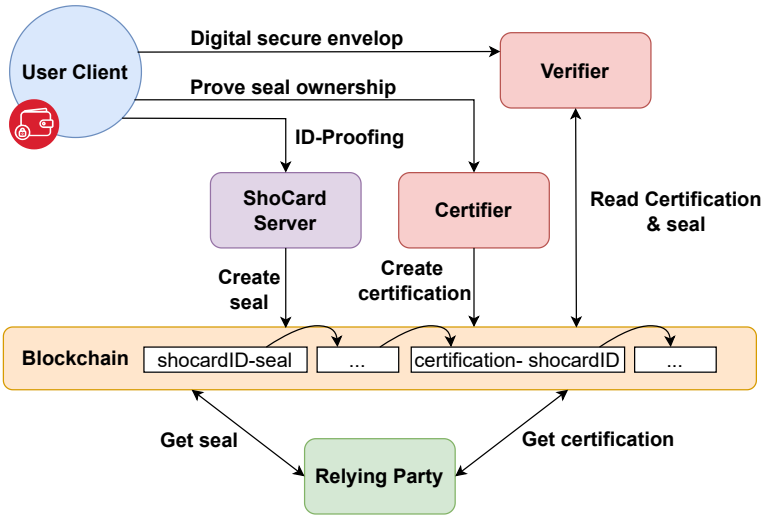


Figure 2.22: Shocard overview

The bootstrapping occurs at the moment of creation of a new Shocard. The user scan their ID documents. The scan and the corresponding data are encrypted and stored on the mobile device; the signed hash of this data is also embedded into a Bitcoin transaction for later data validation purposes. The resulting Bitcoin transaction number constitutes the user's ShocardID and is retained in the mobile application as a pointer to the Shocard seal.

The user can then interact with identity providers to gather additional attributes in a process called certification. To associate certificates with a ShocardID, an identity provider must first verify that the user knows both the hashed data to create it and the cryptographic keys that signed the seal. The certificate takes the form of a signed hash of the new attributes (and their associated ShocardID) in a Bitcoin transaction created by the provider. The provider must share the Bitcoin transaction number along with a signed plaintext of the new attributes directly with the user. Since the user will later need to provide the attributes to relying parties and may not want to lose them if the mobile device is lost, a Shocard server provides storage on which to encrypt the

certificates. Shocard never knows the encryption key, which allows the user to share the certificates only with selected parties.

Finally, the validation phase occurs when a relying party must verify a certification to determine whether a user is entitled to access a service. The user must first provide the relying party with the envelope reference and its encryption key. After retrieving the envelope from Shocard's servers, the relying party performs a series of checks. First, that the signature on the envelope was produced with the same private key that signed the seal; then that the signature on the certification was created by a trusted entity and that the plaintext certification matches the one encrypted and signed on the certification; finally, that the identity data presented by the user in the pending transaction matches the one signed and encrypted on the seal.

Although Shocard is a simple and lightweight solution in which the user has acceptable control over his data, the intermediary role of the server as a central entity makes it, in practice, a much more dependent system than a priori would be desirable. Furthermore, the way in which shocard generates credentials by scanning official documents may mean that users need to enter more data than they would a priori wish in order to generate the certificates. This makes shocard credentials an interesting target for attackers. Moreover, aspects such as unlinkability are not fully guaranteed making it possible to track users.

Hyperledger Aries [61] is a pivotal project within the Hyperledger ecosystem, aimed at developing a toolkit for decentralized identity management using distributed ledger technologies (DLTs). Aries facilitates secure and private interactions among various parties by leveraging blockchain technology to handle digital identity credentials and verifiable claims. Figure 2.23. This solution provides the following key features:

- **Decentralized Identity:** Aries supports the creation, management, and verification of decentralized identities (DIDs) [63]. These identities are self-sovereign, allowing individuals to control their personal data without relying on a central authority.
- **Interoperability:** Aries emphasizes interoperability between different identity systems and DLT networks, providing a common framework for seamless communication and data exchange across diverse systems.
- **Protocols and Standards:** Aries utilizes industry standards such as the W3C Decentralized Identifiers (DIDs) and Verifiable Credentials (VCs) [69]. It supports various communication protocols, including the DIDComm messaging protocol, which facilitates secure and private messaging between parties.
- **Agent-Based Architecture:** Aries employs an agent-based model where "agents" represent digital entities (such as individuals, organizations, or systems) that interact with each other through standardized protocols. These agents manage the issuance, storage, and presentation of credentials.

- **Modularity:** The project offers modular components that can be customized and extended for different use cases, supporting applications from simple credential verification to complex multi-party transactions.

- **Privacy and Security:** Aries incorporates robust security features to protect sensitive data and ensure privacy, including cryptographic techniques for securing communications and ensuring that personal information is only shared with explicit consent.

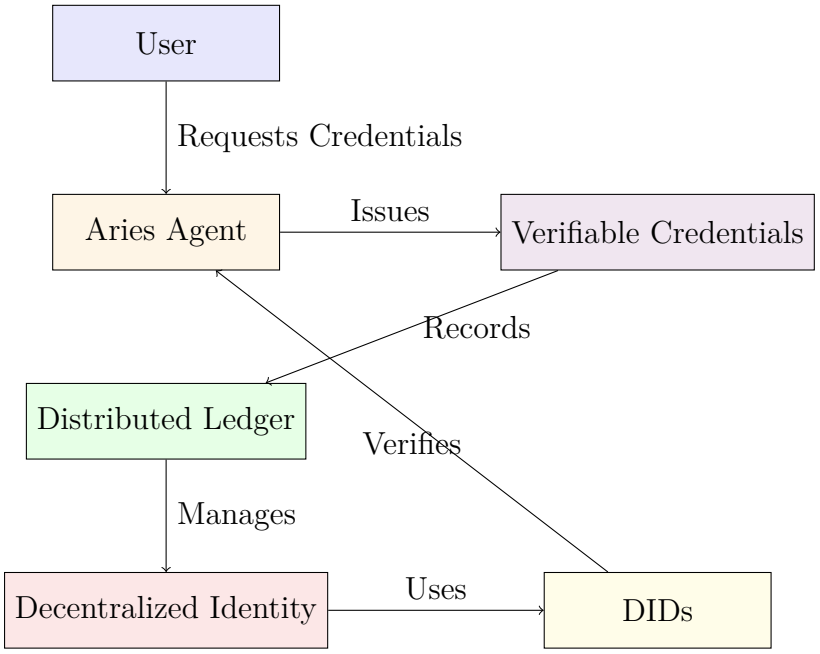


Figure 2.23: Detailed Architecture of Hyperledger Aries

The **User** initiates the process by requesting verifiable credentials from the **Aries Agent**, which is responsible for issuing these credentials and recording them on the **Distributed Ledger**. The Distributed Ledger ensures the secure storage and management of credentials and related identity data. These credentials are utilized by the **Decentralized Identity** framework, which oversees decentralized identity management. **Decentralized Identifiers (DIDs)** play a crucial role in verifying these credentials and interacting with the Aries Agent, thereby validating the accuracy and integrity of identity claims.

In summary, Hyperledger Aries is a robust and evolving toolkit that provides foundational tools for decentralized identity management. Its advanced protocols, modular architecture, and strong privacy protections make it a crucial component of the Hyperledger ecosystem. However, it faces challenges related to complexity, scalability, evolving standards, and adoption that must be addressed to fully realize its potential.

2.3. Innovative identity management projects

In this section, we delve into several innovative projects that have significantly shaped the field of identity management. It's important to recognize the value of both recent and older solutions. While the latest advancements often feature cutting-edge technologies and modern best practices, older solutions provide essential foundational knowledge. By looking at these earlier projects, we can identify persistent challenges, understand how technological solutions have evolved, and appreciate the incremental improvements that have led to today's systems.

Exploring these pioneering efforts helps us grasp the complexities of creating robust, privacy-preserving, and user-friendly identity management frameworks. This historical perspective is crucial as it highlights the ongoing need for innovation and adaptation in response to evolving security threats and privacy concerns. By learning from them we can better address current and future challenges in the field.

2.3.1. ARIES: reliAble euRopean Identity EcoSystem

The European Union's ARIES project [1, 21], an acronym for reliAble euRopean Identity EcoSystem, represents a groundbreaking effort aimed at fortifying the pillars of security, interoperability, and user privacy within electronic identity (eID) systems across Europe. Figure 2.24 provides a comprehensive overview of the ARIES architecture, succinctly illustrating the key components and their interconnections, alongside the processes that underpin the system's operation. This visualization offers a clear, at-a-glance understanding of the project's sophisticated framework, highlighting its innovative approach to enhancing eID systems.

Objectives and Goals

The ARIES project is underpinned by several ambitious objectives, each aimed at enhancing the eID landscape in Europe:

- **Enhanced Security:** ARIES endeavors to fortify eID systems against the prevalent threats of unauthorized access, fraud, and identity theft, ensuring that digital identities are safeguarded with the highest security standards.
- **Interoperability Across Borders:** A cornerstone goal of ARIES is to achieve seamless eID usage throughout the European Union, thereby facilitating a truly integrated digital single market where individuals and businesses can interact across borders with ease and security.
- **Privacy and Trust:** Recognizing the paramount importance of privacy in the digital realm, ARIES commits to implementing advanced security measures that not only protect user data but also foster a climate of trust in eID systems.

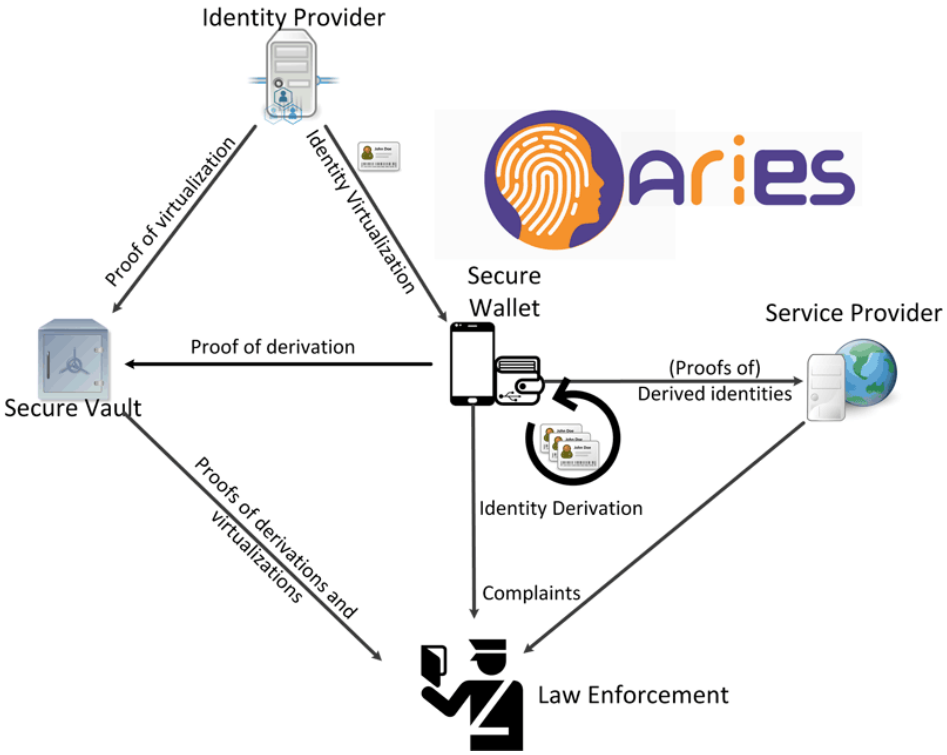


Figure 2.24: ARIES overview [1]

Technological Innovations

ARIES leverages a multifaceted suite of cutting-edge technologies to realize its objectives:

- **Biometric Authentication:** By incorporating biometric verification methods such as fingerprint scanning, facial recognition, and iris scans, ARIES introduces a highly secure and user-friendly mechanism for identity verification. This approach minimizes the risk of identity theft and fraud, ensuring that eID systems are accessible solely by their legitimate owners.
- **Blockchain and Distributed Ledger Technologies (DLT):** Utilizing blockchain technology, ARIES aims to revolutionize identity management with a decentralized framework. This innovation not only enhances the security and transparency of eID systems but also establishes a tamper-proof record of identity transactions, offering a new level of trust and integrity in digital identities.
- **Modern Cryptography:** At the heart of the ARIES security model is the use of advanced cryptographic techniques, including public key infrastructure (PKI) and secure multi-party computation. These technologies are crucial for protecting data during transmission and storage, ensuring the privacy and security of personal information within the ARIES ecosystem.

Challenges and Solutions

Despite its innovative approach, the ARIES project encounters several significant challenges:

- **Balancing Security with Usability:** A key challenge is to design a system that is both highly secure and easily accessible to users. ARIES addresses this by prioritizing user-centric design principles, ensuring that enhanced security measures do not compromise the user experience.

- **Privacy Preservation:** In an era of increasing digital surveillance, maintaining user privacy is a formidable challenge. ARIES tackles this by implementing privacy-enhancing technologies and data minimization practices, thereby safeguarding personal information against unauthorized access and exploitation.

- **Interoperability:** Achieving interoperability across diverse technological platforms and jurisdictions is a complex task. ARIES meets this challenge through the development of standardized protocols and interfaces that ensure seamless integration of eID systems across the European Union.

Impact and Relevance

The ARIES project is poised to make a substantial impact on the European digital economy:

- **Strengthening Digital Services:** By providing a more secure and interoperable framework for eIDs, ARIES has the potential to significantly enhance the delivery of digital services across Europe, benefiting both public services and private enterprises.

- **Facilitating the Digital Single Market:** ARIES supports the European Union's vision of a digital single market by enabling more efficient and secure cross-border transactions, thus driving economic growth and innovation across member states.

- **Building Trust in Digital Transactions:** Through its focus on security and privacy, ARIES plays a vital role in building trust in online transactions, a crucial factor for the expansion of the digital economy.

Drawbacks and Limitations

While the ARIES project represents a significant advancement in the realm of digital identity management, it is not without its drawbacks and limitations:

- **Technological Complexity:** The sophisticated technologies employed by ARIES, while innovative, also introduce complexity that could hinder widespread adoption and integration, particularly among smaller organizations and individuals with limited technical expertise.
- **Privacy Concerns:** Despite efforts to enhance privacy, the centralization of sensitive biometric data and the potential for surveillance raise concerns about the long-term implications for user privacy and autonomy.
- **Interoperability Challenges:** While ARIES aims to achieve interoperability across EU member states, differing national regulations, and technological standards may limit the project's effectiveness and scalability.

Conclusions

The ARIES project marks a pivotal step toward establishing a more secure, interoperable, and privacy-conscious digital identity ecosystem in Europe. Its innovative approach, leveraging state-of-the-art technologies, addresses many of the current challenges faced by eID systems. However, the project's success will ultimately depend on its ability to navigate the inherent trade-offs between security, usability, and privacy. As ARIES moves forward, it will be crucial to continually assess and address these drawbacks and limitations to fully realize the vision of a unified and secure digital Europe.

2.3.2. ABC4Trust

The ABC4Trust project [2], an acronym for 'Attribute-based Credentials for Trust,' is an ambitious project to redefine the landscape of digital identity and privacy. This initiative supported by the European Union, strives to forge a cohesive framework for the deployment of Privacy-ABCs (Attribute-Based Credentials) [12]. These credentials stand at the forefront of privacy enhancement technologies, designed to elevate privacy and trust across the digital realm. Privacy-ABCs empower users to disclose solely the indispensable information required for online transactions. This minimalist approach to data sharing significantly mitigates the risk of personal data overexposure, thereby cultivating a digital environment where security and privacy are paramount.

Figure 2.25 unveils the architecture of the ABC4Trust framework, providing an insightful depiction of its foundational components and their interactions. Also elucidates the operational processes that constitute the backbone of the system.

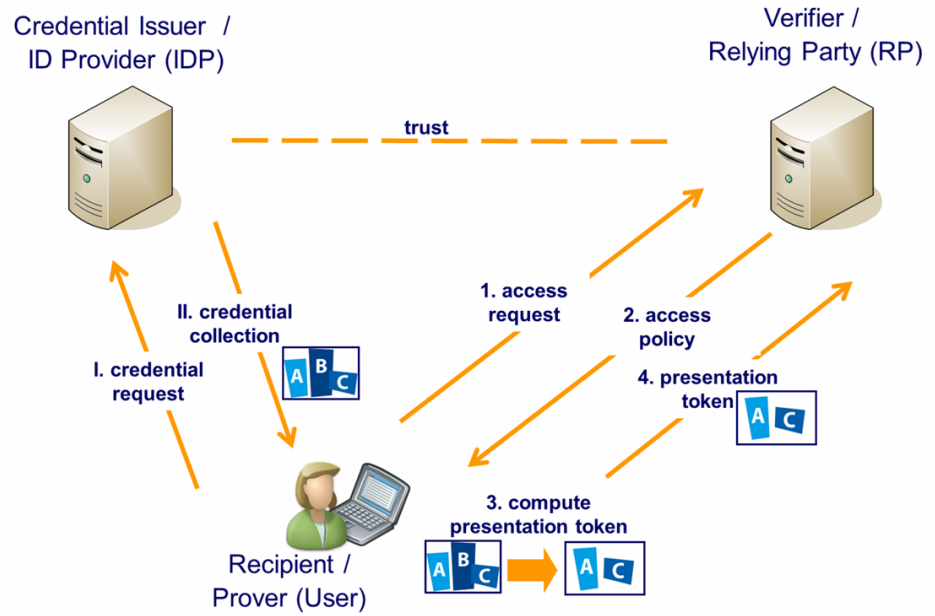


Figure 2.25: ABC4Trust overview [2]

Objectives

The overarching objectives of ABC4Trust are multifaceted and ambitious, aiming to address key challenges in digital privacy:

- **Framework Establishment:** To create a unified and interoperable framework for Privacy-ABC systems that can seamlessly integrate across diverse platforms and services, promoting widespread adoption.

- **Selective Disclosure:** To empower users with the capability to precisely control what personal information is disclosed during online interactions, thereby enhancing individual privacy.
- **Trust Enhancement:** To bolster trust in digital ecosystems through the provision of secure and verifiable transactions, thus facilitating a safer online environment.
- **Practical Demonstrations:** To showcase the viability and benefits of Privacy-ABCs through pilot projects in real-world settings, thereby illustrating their practical applicability and impact.

Technological Innovations

ABC4Trust introduces several technological advancements aimed at redefining digital identity verification:

- **Advanced Cryptography:** The project employs sophisticated cryptographic mechanisms to enable authentication without the disclosure of excessive personal data, prioritizing user privacy without compromising security.
- **User-Centric Credentials:** ABC4Trust's approach to credentials allows for the association of user attributes with verification processes without directly linking these attributes to the user's identity, thereby supporting anonymity and reducing traceability.
- **System Interoperability:** A key innovation of the project is the development of a framework that ensures the compatibility of Privacy-ABCs across different technological platforms and services, thus addressing a critical barrier to adoption.

Challenges and Solutions

While ABC4Trust's approach presents a compelling vision, it encounters several significant challenges:

- **User Adoption:** The success of Privacy-ABCs hinges on widespread user acceptance and understanding. To this end, ABC4Trust emphasizes the importance of user education and the simplification of the user experience to facilitate adoption.
- **Balancing Act:** Achieving a balance between robust security measures and user-friendly authentication processes is crucial. The project advocates for a design philosophy that does not sacrifice usability for the sake of enhanced security.
- **Regulatory and Technical Interoperability:** Navigating the complex landscape of varying regulatory requirements and technological standards poses a significant challenge. ABC4Trust addresses this through active collaboration with stakeholders, regulatory bodies, and technology providers to develop standardized protocols and interfaces.

Impact and Relevance

The potential impacts of the ABC4Trust project are profound and far-reaching:

- **Empowering Users:** By granting users greater control over their personal data, ABC4Trust sets the stage for a shift towards more privacy-respecting digital services, thereby fostering trust and confidence in online transactions.
- **Promoting Secure Transactions:** The project's emphasis on secure and verifiable transactions paves the way for advancements in e-commerce, e-government, and beyond, promoting a safer online environment for all stakeholders.
- **Privacy-Enhancing Precedent:** ABC4Trust's successful implementation could serve as a model for future digital identity systems, demonstrating the viability of privacy-enhancing technologies in mainstream applications.

Drawbacks and Limitations

Despite its innovative approach, the ABC4Trust project is not without its drawbacks and limitations:

- **Implementation Hurdles:** The complexity of the technologies involved and the need for infrastructural changes pose significant challenges to the rapid adoption of Privacy-ABCs, particularly among smaller entities and those with limited technological resources.
- **Data Collection Practices:** Resistance from entities that benefit from extensive data collection and processing may impede the project's progress and broader acceptance.
- **User Education:** The need for extensive user education to convey the benefits and operations of Privacy-ABCs represents a considerable challenge, necessitating dedicated efforts to raise awareness and understanding.

Conclusions

The ABC4Trust project embodies a pivotal advancement towards establishing a privacy-centric paradigm in digital identity management. By championing the cause of Attribute-Based Credentials, the project not only aims to safeguard privacy and enhance trust in online transactions but also to lay the groundwork for a future where users possess unequivocal control over their personal information.

2.3.3. PrimeLife

The European Union's groundbreaking initiative, "Privacy and Identity Management in Europe for Life" (PrimeLife) [31], represents a significant effort to tackle the intricate challenges associated with privacy and identity management within the vast expanse of the digital world. Anchored firmly in the European Union's enduring commitment to data protection and individual privacy rights, the PrimeLife project is driven by a vision to innovate and implement a comprehensive suite of technologies and methodologies. These advancements are designed to offer robust protection for personal information, effectively countering the dynamic and increasingly sophisticated array of threats that pervade the online environment. PrimeLife's objectives extend beyond the technological realm, aiming to influence policy and regulatory frameworks within the European Union. By aligning its technological advancements with the EU's legal standards, including the General Data Protection Regulation (GDPR) [3], PrimeLife endeavors to shape future policies that foster a balance between innovation and privacy protection.

Objectives

At its core, PrimeLife's objectives are both visionary and practical, focusing on:

- **Comprehensive Framework Development:** Crafting a robust framework that encompasses the entire lifecycle of identity management, from creation to deletion, ensuring that privacy remains a cornerstone throughout.
- **Empowerment through Tools:** Providing citizens with powerful, yet intuitive, tools that offer unprecedented control over their personal data, enabling them to navigate the digital world with confidence.
- **Cross-Border Interoperability:** Bridging the gap between diverse identity management systems across Europe, PrimeLife seeks to create a seamless experience for users engaging in cross-border activities, thereby enhancing the Digital Single Market.
- **Policy Influence:** Actively contributing to the dialogue around privacy and data protection policies, PrimeLife aims to influence and shape future legislation to reflect its findings and technologies.

Technological Innovations

PrimeLife introduces a range of technological innovations designed to enhance online privacy and identity management:

- **Cutting-Edge Privacy-Enhancing Technologies (PETs):** PrimeLife incorporates advanced PETs to ensure data privacy and security. These include:

- *Attribute-Based Credentials (ABCs) [12]:* Allowing verification of attributes without revealing user identity or unnecessary information.
- *Homomorphic Encryption [70]:* Enabling computations on encrypted data, resulting in encrypted outputs that, once decrypted, reveal the desired outcome without compromising privacy.
- *Zero-Knowledge Proofs (ZKPs) [71]:* Facilitating the proof of truth of a statement without revealing any information beyond the validity of the statement itself.
- *Secure Multi-Party Computation (SMPC) [72]:* Permitting joint computations on private inputs, producing a result without exposing those inputs to others.
- *Anonymous Credentials [43]:* Supporting privacy-preserving authentication by allowing users to prove credentials without disclosing identity.
- *Decentralized Identifiers (DIDs) [63]:* Enabling verifiable, self-sovereign identities that do not rely on central authorities, enhancing privacy and control over personal data.

- **User-Centric Identity Management Tools:** Developing intuitive tools that empower users to manage their digital identities, with features that prioritize user consent and minimal data exposure.
- **Blockchain for Enhanced Privacy:** Exploring blockchain technology to create secure, decentralized systems for identity and data management, ensuring integrity and transparency while maintaining user privacy.

These technological solutions collectively aim to address the challenges of privacy and identity management, providing secure and user-friendly mechanisms for individuals to navigate the digital world with confidence.

Challenges and Solutions

Navigating the challenges inherent in such a pioneering project, PrimeLife proposes several solutions:

- **Ensuring Broad User Adoption:** Through extensive outreach, education, and by demonstrating the tangible benefits of the PrimeLife tools, the project seeks to encourage widespread adoption.
- **Addressing Technical Complexity:** By fostering collaboration among leading technologists, developers, and researchers, PrimeLife aims to simplify the integration of PETs into existing infrastructures.
- **Staying Ahead of Regulatory Changes:** PrimeLife remains agile, ready to adapt its technologies and frameworks in response to new regulatory requirements, particularly those emerging from the GDPR.

Impact and Relevance

The implications of PrimeLife's success are profound:

- **Redefining Online Privacy:** By setting new standards for privacy and data protection, PrimeLife has the potential to drastically alter how personal information is managed online, fostering a safer digital environment.
- **Influencing Future Technology Development:** As PrimeLife integrates and demonstrates the effectiveness of PETs, it paves the way for their adoption in future digital services and platforms.
- **Shaping Policy and Legislation:** The project's insights and technologies are poised to influence European privacy legislation, ensuring that future policies are grounded in practical, tested solutions.

Drawbacks and Limitations

While PrimeLife's ambitions are commendable, the project faces several hurdles:

- **Technological Adoption and Integration:** The broad implementation of PrimeLife's outcomes may encounter resistance due to the complexity of its technologies and the need for substantial infrastructural changes.
- **Balancing Privacy with Usability:** Crafting solutions that enhance privacy without compromising on user experience remains a delicate balancing act.
- **Sustaining Long-Term Engagement:** Ensuring the continued relevance and adoption of PrimeLife's innovations requires ongoing support, updates, and community engagement.

Conclusion

The PrimeLife project embodies the European vision for a digital future anchored in privacy, security, and user empowerment. Through its groundbreaking technologies and commitment to user-centric solutions, PrimeLife seeks not only to navigate the challenges of digital identity management but to redefine them, offering a blueprint for a safer, more private online world.

2.4. Conclusions

In this chapter, we have reviewed the background and state of the art in identity management systems, with a focus on enhanced privacy and distributed ledger technologies (DLT). The examination of both traditional and innovative identity management projects has provided a comprehensive understanding of the current landscape.

Conclusions

Several conclusions can be drawn from this review:

- **Evolution of Identity Management:** Traditional identity management systems, such as those based on centralized models (e.g., OpenID, OAuth, SAML), have laid the groundwork for managing digital identities. These systems facilitate user authentication and authorization across various services, offering convenience through Single Sign-On (SSO) mechanisms. However, they also suffer from significant privacy and security limitations, including the centralization of trust, which makes them vulnerable to data breaches and single points of failure. This aligns with our objective to analyze current identity management systems and identify key challenges (**O1**).
- **Emergence of DLT:** Distributed ledger technologies, particularly blockchain, offer promising solutions to the challenges posed by traditional identity management systems. By decentralizing trust and enhancing transparency, DLT can address issues related to data integrity, privacy, and security. These technologies facilitate the creation of tamper-proof records and enhance the robustness of identity verification processes. This insight supports our objective to investigate the application of DLT in identity management systems (**O2**).
- **Innovative Projects:** Projects like ARIES, ABC4Trust, and PrimeLife have demonstrated the potential for privacy-preserving identity management solutions. ARIES, for instance, integrates biometric authentication and secure elements to ensure high levels of assurance. ABC4Trust focuses on attribute-based credentials to enhance privacy, while PrimeLife explores life-long privacy protection. Despite their varying degrees of success and adoption, these initiatives have contributed valuable insights and technological advancements to the field. This finding is closely related to our objective to design a solution for identity management applying distributed technologies (**O4**).
- **Self-Sovereign Identity (SSI):** The move towards self-sovereign identity models, where users have greater control over their personal data, represents a significant shift in the identity management paradigm. SSI models empower users to manage their identities independently, reducing reliance on centralized identity providers. This approach aligns well with regulatory requirements like GDPR and addresses many of the shortcomings of traditional systems by minimizing data collection and enhancing user privacy. This directly relates to our objective to combine distributed identity management with DLT to enhance privacy and trust (**O5**).

GAP Analysis

Despite the progress made, several gaps remain in the current identity management landscape:

- **Integration with Existing Systems:** One of the major challenges is the seamless integration of new identity management solutions with existing systems. Ensuring compatibility with widely used standards (e.g., OpenID, OAuth, SAML) is crucial for widespread adoption. Many organizations have already invested heavily in existing infrastructures, and any new solution must offer a clear path for integration without requiring a complete overhaul of current systems. This is important for verifying the obtained identity solutions in real scenarios (**O6**).
- **User Adoption and Usability:** Advanced identity management solutions must balance security and privacy features with user-friendliness. High complexity and poor usability can hinder adoption, even if the solutions offer superior security and privacy. User education and intuitive interfaces are essential to encourage widespread acceptance and use. Additionally, users must trust the system, which means transparent communication about how their data is managed and protected. This aligns with the objective to analyze current identity management systems to identify key challenges (**O1**).
- **Scalability and Performance:** Distributed ledger technologies, while promising, still face challenges related to scalability and performance. Ensuring that these systems can handle large-scale deployments without compromising on speed or efficiency is essential. Blockchain, for example, can suffer from latency and high resource consumption, which must be addressed to make it viable for mainstream identity management applications. This relates to our objective to investigate the application of DLT in identity management systems (**O2**).
- **Regulatory Compliance:** Ensuring compliance with evolving data protection regulations remains a critical challenge. Identity management solutions must be designed with regulatory requirements in mind to avoid legal issues and ensure user trust. Regulations such as GDPR impose stringent requirements on data handling, consent management, and user rights, necessitating that new systems be built with these considerations at their core. This is connected to our objective to combine distributed identity management with DLT to enhance privacy and trust (**O5**).
- **Trust Management:** Enhancing trust among users, service providers, and identity providers is a persistent challenge. Developing robust mechanisms for trust verification and management is crucial for the success of decentralized identity systems. Trust frameworks must be established to ensure that all parties involved in identity transactions can be reliably authenticated and authorized. This relates to our objective to analyze the main current implementations of DLT technologies to learn about their strengths and limitations (**O3**).
- **Privacy and Security Trade-offs:** Achieving a balance between privacy and security is a delicate task. While enhancing privacy is a priority, it should not compromise the security of the identity management system. Solutions must

ensure that privacy-preserving techniques, such as zero-knowledge proofs and homomorphic encryption, do not introduce vulnerabilities or reduce the overall security posture. This supports our objective to design a solution for identity management applying distributed technologies (**O4**).

- **Interoperability:** The lack of standardization and interoperability between different identity management systems and technologies can create barriers to adoption. Ensuring that new solutions can work seamlessly with various platforms, services, and protocols is essential for creating a cohesive and functional identity management ecosystem. This is crucial for verifying the obtained identity solutions in real scenarios (**O6**).

- **Cost and Resource Allocation:** Implementing advanced identity management solutions can be resource-intensive, requiring significant investment in technology, training, and maintenance. Cost considerations can be a barrier for smaller organizations or those with limited budgets. Solutions must be designed to be cost-effective and scalable, offering flexible deployment options to suit different organizational needs. This aligns with our objective to analyze current identity management systems to identify key challenges (**O1**).

Addressing these gaps will require ongoing research and innovation. By building on the successes and learning from the limitations of current and past projects, the field of identity management can continue to evolve towards more secure, privacy-preserving, and user-friendly solutions. Future work should focus on developing integrated, scalable, and compliant identity management frameworks that can adapt to the dynamic landscape of digital identities.

CHAPTER 3

Privacy-preserving distributed identity management

3.1. Introduction

Personal data has become the new critical element to manage, protect and, of course, compromise. The rise of smart cities, e-health, the new Industry 4.0 and growing cloud services are challenging traditional identity management systems, which are not evolving as fast as desirable. In addition, other AI-based technologies and their systematic analysis of data, the reduction of storage costs and the scarcity of tools that allow users to manage their private data pose a serious problem for consumers and an advantage for identity and service providers. Data such as location and health data are of great value to companies, which often collect them even without users' knowledge. Traditional identity management systems (IdM) are focused on the use of centralised identity providers (IdPs) that create, manage, and maintain the identity information of their users or smart devices while providing mechanisms for authentication to different service providers. An example of this would be Google, which allows us as users to log in to a wide variety of third-party services acting as an identity provider. This is a widespread solution and while it is convenient and simple to operate, achieving certain levels of security and privacy is a challenge. Tracking and binding by IdPs is one of the main issues to be addressed. For example, in applications dealing with sensitive data (e.g., health), loss of privacy can be a major issue.

In this context, users need to be cautious about how, when and with whom they share their personal information, in order to minimize the risk of data leakage or collection without consent [73, 74]. In addition, they should be provided with the necessary tools to enable them to effectively exercise the rights described by regulations such as the

General Data Protection Regulation (GDPR) [3, 75] implemented by the European Union or other similar directives. Current authentication and IdM mechanisms struggle to meet security and privacy requirements while maintaining usability levels. At best, websites or service providers verify email addresses and phone numbers by sending one-time codes. Age verification, which should be a common use case given the amount of age-restricted material offered online, is often done by verifying a credit card number, even though credit cards were never intended for this purpose.

The issuance of electronic ID cards in several countries has been an attempt to improve the situation, however, these IDs are often smart cards that suffer from impractical usability when combined with devices such as smartphones, tablets or computers. In addition, there is poor cross-country compatibility, forcing service providers to choose which cards to support and which not to support. As a result, the classic username and password mechanism remains the most widespread and popular way to authenticate online despite the difficulties of remembering different identities on different services or even taking the risk of reusing passwords between different services.

The username and password system has only been improved in terms of usability by the introduction of SSO systems which have allowed the centralization of identity management for several services through a single username and password combination. On the other hand, the gain in usability has penalized privacy and security features by introducing a critical failure point in the system. The central IdP is involved in every authentication performed for a service provider, becoming a Big Brother that is able to track user habits, correlate accounts between different services, expose private data if compromised and even impersonate the user's identity.

To address these shortcomings, this thesis proposes a privacy-preserving identity management solution by applying a distributed cryptographic approach to currently deployed identity management technologies. The proposed solution splits the role of the traditional identity provider (IdP) among multiple partial identity providers, so that none of them alone can impersonate or track its users. In addition, the solution aims to be integrable with other existing technologies and standards, reducing hardware requirements and offering a user recognisable functionality based on username and password.

3.2. Concept

Current identity management solutions can be divided into two types. On the one hand, (1) online solutions (e.g., SAML, OpenID Connect, etc.), where the identity provider is actively involved in the authentication process. A user who wants to access a service is redirected to the identity provider (IdP), which performs the authentication process (e.g. by username and password) and produces a short-lived access token that can be verified by the service provider (SP). During this process, the IdP acts like a "Big brother" that knows all the details of all access requests made by users, enabling user tracking and compromising their privacy. Moreover, a malicious IdP is able to

impersonate any of its users and because of that it becomes a very attractive target for attackers.

On the other hand, (2) offline solutions, where the IdP is only involved only during the process of issuing a long-lived credential (e.g. X.509 certificates [30] etc). The user contacts the IdP to authenticate and obtain a credential (issuing process) that can be stored for later use by generating access tokens to access services without the IdP having to intervene. Unlike online, the offline process does not suffer from the same privacy issues. This system reduces user exposure by issuing credentials to an isolated system (e.g. a wallet on the user's device). Nevertheless, the main problem with this solution is the need to delegate the handling of sensitive material to users, who must securely store their long-lived credentials, making them a direct target for attackers. In addition, most of these systems require specific hardware to function properly (e.g., smart cards) or user-unfriendly software extensions.

The approach seeks to evolve the traditional systems to a distributed system, figure 3.1. Behind this evolution are two clear goals: (1) to prevent IdPs from abusing their power to track users and (2) to eliminate the critical point of failure they currently pose. To achieve these goals, we propose the application of distributed cryptography techniques so that the role of the central IdP evolves towards a distributed architecture.

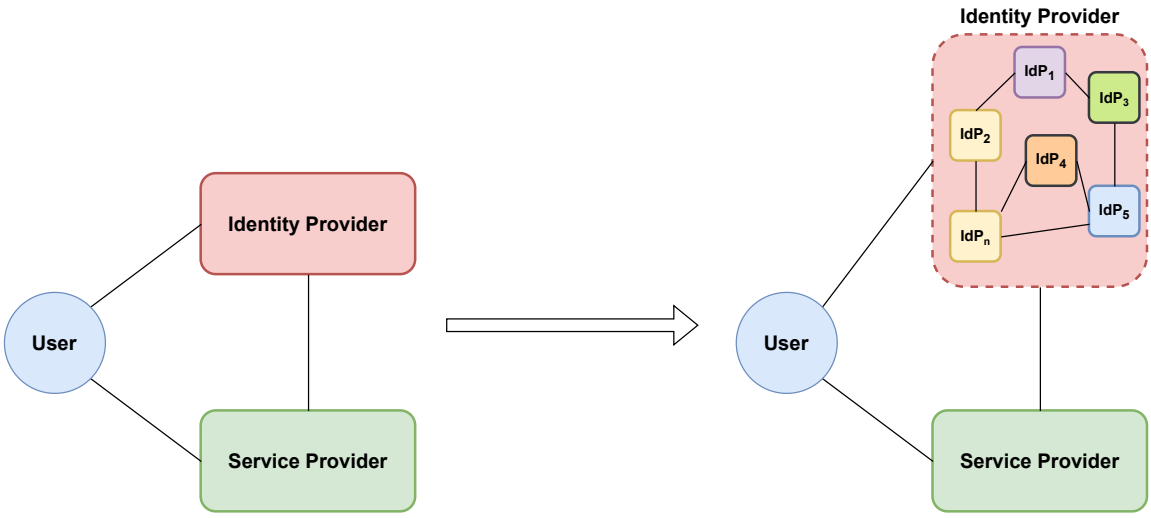


Figure 3.1: Conceptual idea

Distributing the role of the IdP is not a simple task as it has implications for the security and privacy of the users. Firstly, a group of corrupted or compromised IdPs may not be able to issue valid cryptographic material. That is, unless all IdPs are under the control of an attacker, it will not be possible to impersonate a user of the system. With respect to privacy, the system must provide data minimisation, unlinkability and untraceability features. The material presented to service providers should only contain the minimum necessary information. Regarding unlinkability, two or more access tokens cannot be linked together, preventing a service provider or even a coalition of service

providers from being able to eavesdrop on user behaviour. Finally, IdPs should limit themselves to generating authentication material. That is, IdPs will know that a user is performing an authentication process, but they will not learn which service provider the user is contacting.

3.3. Objectives and requirements

Chapter 1 of this thesis presents a set of general objectives and requirements together with a set of specific objectives described in section 1.3. This chapter covers objectives **O4** and **O6**.

The first objective **O1** established includes the need to study and analyse the restrictions present in the current identity management systems, as well as to obtain a list of problems that must be considered in the proposed solution. In this regard, the following key points have been identified:

- Existing solutions are not well balanced between usability and privacy protection.
- In SSO solutions, IdP is a very dominant element. Resulting in loss of privacy through user-tracking techniques across services.
- Compromising an identity provider in SSO systems is catastrophic, putting the all the user data in risk.
- Credential-based solutions are not usable enough to achieve good adoption.
- Users have poor control over how their data is used.
- There is a wide variety of devices that must work correctly with the chosen solution.
- Users do not want to change the mechanisms they already know for unknown ones.

Having identified the gaps in traditional IdM solutions, the fourth objective **O4** focuses on developing an identity management solution that at least maintains the same levels of security while integrating distributed technologies. In that sense, the proposed solution should address the following challenges:

Challenge 1 To establish an identity management system that ensures secure and privacy-friendly identity management interactions.

Challenge 2 Reduce or eliminate the critical point of failure that centralised IdPs represents in SSO solutions while maintaining usability.

Challenge 3 Prevent the IdP from tracking or impersonating its users.

Challenge 4 Support different scenarios to enable users to use different identities when accessing different online or offline services.

Challenge 5 Use well-known authentication mechanisms to reduce the impact over users adoption such as the user-password technique.

Challenge 6 Keep minimum requirements low in order to maximise the number of devices that can support the solution.

Challenge 7 Provide better user control over their data and how it is shared.

In order to achieve the above objectives, the solution must meet a number of requirements that are classified between security and usability. The security requirements determine the minimum features needed to make the solution at least as secure as traditional solutions and add new points for the new security features we have set out in the previous objectives.

RQ.ID	**Name**	**Description**
se.RQ.1	No impersonation by IdPs	A coalition of fewer than a threshold number of IdPs cannot impersonate the user.
se.RQ.2	Hiding SPs from IdP	IdPs cannot observe which user is accessing which SP.
se.RQ.3	Authentication	All components and entities must use mutual authentication protocols.
se.RQ.4	Data integrity	Components must ensure data integrity during communications.
se.RQ.5	Confidentiality	All components must maintain data confidentiality.
se.RQ.6	Availability	The unavailability of any component must not compromise security.
se.RQ.7	Access control	Components must enforce access rules to ensure only authorized entities access sensitive data.
se.RQ.8	Replay protection	All cryptographic protocols must resist replay attacks.
se.RQ.9	Token standards	Authentication tokens should comply with relevant standards, cryptography requirements, and protection profiles.
se.RQ.10	Token authenticity	Tokens must accurately reflect what the vIdP asserts.
se.RQ.11	Proactive security	IdP secret key material can be refreshed to enhance security.

Table 3.1: Security and privacy requirements

As for usability requirements, these should ensure that the proposed solution is accessible to users in order to achieve good adoption.

RQ.ID	Name	Description
us.RQ.1	Effectiveness	All users should be able to authenticate themselves and make use of the proposed solution without previous special knowledge or training.
us.RQ.2	Efficiency	Users must have the perception that the time spent on the whole authentication process is acceptable.
us.RQ.3	Satisfaction	The authentication process should be an experience that meets the user's expectations, i.e. not tedious, time-consuming or labyrinthine.
us.RQ.4	Interoperability	Different scenarios and use cases require different authentication needs. The system must be flexible enough to accommodate as many scenarios as possible.
us.RQ.5	Mobile support	The use of mobile devices for authentication is already a fundamental part of both users and service providers. The solution must work properly on these devices.

Table 3.2: Usability requirements

In the following sections we will discuss the basic building blocks applied to the solution as well as the processes and the proposed architecture. This will be followed by an evaluation through a set of use cases and finally the main results obtained.

3.4. Processes and architecture

3.4.1. Overview

The proliferation of digital services necessitates robust identity management systems that safeguard user privacy while ensuring seamless access to services. Traditional centralized identity providers, while widely adopted, present challenges including single points of failure and privacy concerns. Now, a novel approach to identity management through a Distributed Identity Protocol (DIP) is introduced, leveraging a Virtual Identity Provider (vIdP) to distribute the authentication mechanism across multiple entities, enhancing security and user privacy.

Figure 3.2 presents an evolution for identity management from a centralised to a distributed model in which the identity provider is no longer a single monolithic entity and becomes a distributed one. Through the collaboration of different partial identity providers, a logical entity called *Virtual Identity Provider* (vIdP) is introduced.

Unlike traditional solutions, the vIdP is not a single point of failure, at least half plus one of the partial IdPs need to be compromised to jeopardise the infrastructure. To achieve the distributed model, the solution relies on cryptographic techniques based on threshold cryptography [76] and to increase privacy features, it includes anonymous credentials [77] and secure multi-party computing [78] to ensure that none of the partial IdPs or a coalition, is able to track or impersonate users.

The proposed architecture for the **Distributed Identity Protocol (DIP)** consists in three main entities: (1) **User**, (2) **Service Provider** and (3) **Virtual IdP**.

1. **User**: The user actively interacts with both the virtual IdP and the service provider. The user needs to authenticate with the virtual identity provider in order to obtain authentication material (token). The authentication of the user is carried out by a typical username-password pair and once authenticated, the user obtains the access token which is presented to the corresponding service which verifies the its validity.

2. **Service Provider (SP)**: It protects access to a set of resources or services. It establishes access policies that the user must satisfy in order to gain access and also verifies that the access token presented to it by the user is valid.

3. **Virtual IdP**: It consists of a set of identity providers (hereafter partial IdPs) which do not have to be fully trusted and which, using cryptographic techniques, collaborate and behave as a single identity provider. The vIdP provides distributed authentication and issuance mechanisms in such a way that all partial IdPs must participate in both processes by verifying user's access data (username and password) as well as during the issuance process.

The proposal supports two types of independent scenarios, online and offline. In the online case, the user experience is identical to that of any existing SSO [26–28, 34] system. The user is redirected to the vIdP when tries to access a service provider to authenticate via username and password. Finally, after a successful process the user is redirected to the service with an authentication token. This token includes the minimum information necessary to satisfy the access policy and additionally, the user may be asked to actively participate either to agree to the disclosure of certain information (i.e, age, nationality ...) or for information purposes only. As for the second mode of operation, the offline case, it ensures that the user may be able to access a service provider even when there is no connectivity to the vIdP. This scenario involves the issuance of credentials that are stored in a digital wallet. The wallet is a piece of software (i.e., an Android app) that does not require specific hardware to function and making it simpler to adopt than smart cards. Therefore, a user is able to authenticate against the vIdP to obtain a credential for later use. As with solutions such as X.509 [35], the credential is a critical material to protect and the responsibility lies with the user.

The combination of threshold cryptography together with secure multi-party computing prevents partial IdPs from being able to know pseudonyms with which users

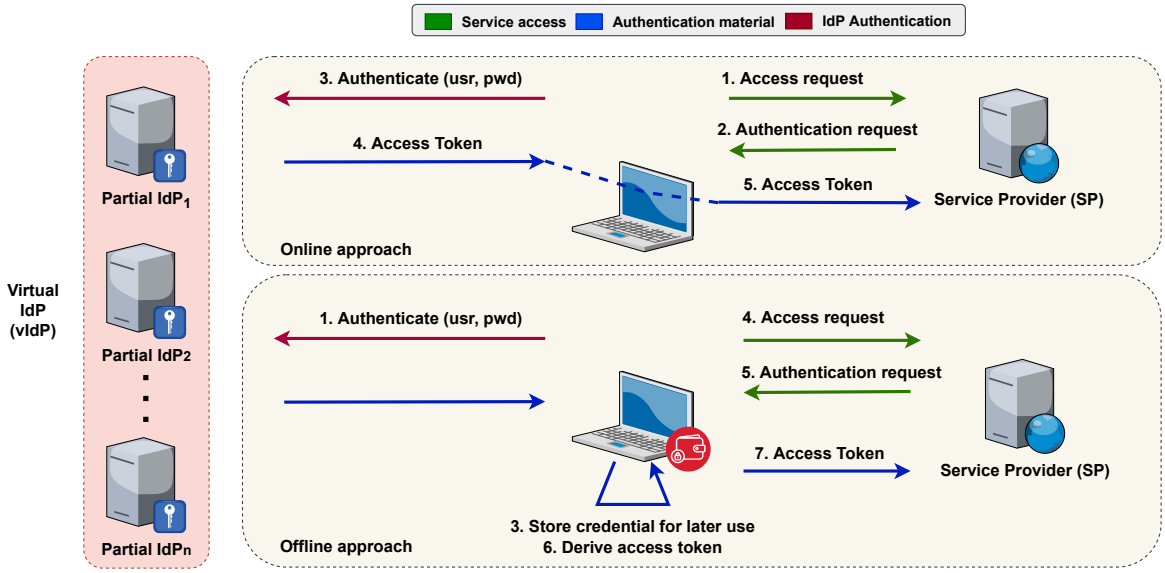


Figure 3.2: Overview of the distributed identity management system

access service providers. In addition, it prevents linking different authentication requests from one user, not being able to know whether they were for the same SP or for a different one. Moreover, even in the case where an SP and a coalition of partial IdPs collate their records, they will not be able to relate the authentication tokens received by the SP to the user's session.

From the perspective of users and service providers, the solution is fully transparent. The user interacts with the vIdP as usual, as if it were a traditional IdP. This facilitates the acceptance of the solution, as nothing seems to change. Neither does the service provider see any change in traditional behaviour. The proposed scheme also maintains the way in which users authenticate themselves in the IdP, using the typical username and password pair that everyone knows. However, the decision to maintain a traditional authentication system, makes password protection crucial. The main problem with password-based systems is that if the IdP where the password is stored is compromised, the security disappears. In the best case scenario, the attacker will only have a database with usernames and salted-passwords that would allow him to perform offline attacks, e.g. using dictionaries. In a catastrophic case, the passwords are stored unprotected and the attacker gains access to all accounts immediately. This approach cannot afford to have the password replicated in N partial identity providers. It is vital to prevent partial IdPs from learning or storing the password. This solution provides a first major improvement over traditional systems: A **distributed password verification (DPV)**, figure 3.3. Instead of storing the password, each partial IdP stores only a portion of a secret key, used to compute a pseudorandom function operation with the user password preventing any of the partial IdPs from knowing a user's real password at any time and by actively cooperating in the authentication process. Only through the cooperation is it possible to verify the validity of a password.

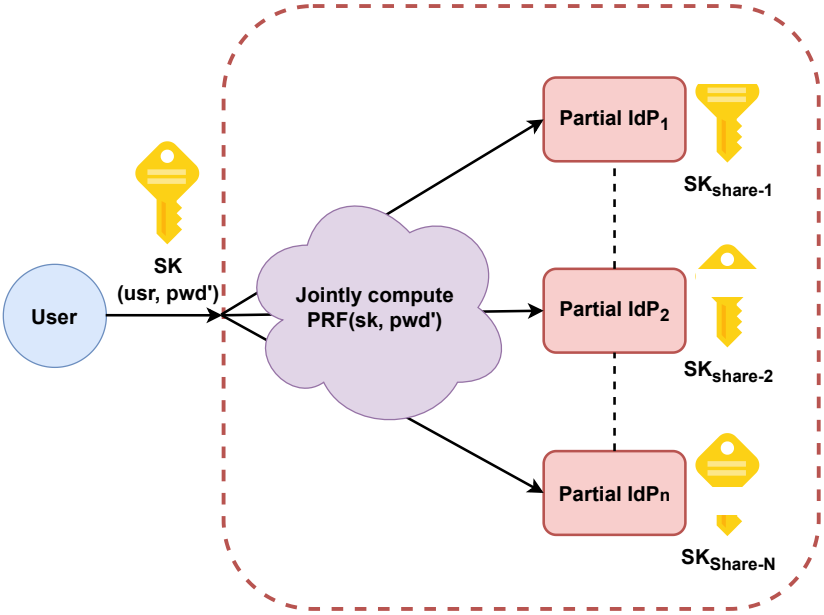


Figure 3.3: Distributed password high level

In order for the solution to support online and offline scenarios, the vIdP must be able to generate valid cryptographic material in a distributed manner. For this purpose, the solution has two procedures (1) **Distributed token generation (DTG)** and (2) **distributed credential issuance (DCI)**. Both processes are independent and do not require each other, however, both are protected by the **DPV** process. The figure 3.4 shows a detailed view of how the different components and processes interact.

3.4.2. Architecture

The main idea is to split the functionality into interoperable modules in order to facilitate the development and integration process. In this way, each partial identity provider includes three modules that are responsible for the distributed authentication of users and the distributed issuance of tokens or credentials, depending on whether it is an online or offline scenario. On the other hand, a user who wants to operate with DIP must also integrate the corresponding modules for authentication and for token or credential handling (typically, a simple Android or iOS app). Finally, the service provider will only need to add to its verifier stack (e.g. Google, Facebook...) the verifier module of the DIP solution.

- **Distributed Password Verification (DPV)**: IdPs store login information for each user who wishes to use the system. Only by knowing the password, a user will be able to successfully log into the system. Password verification is distributed, which means that all partial IdPs must participate in this process otherwise the task will not be completed. More precisely, a set of IdPs register a user by storing

account information (e.g. associated attributes). At each user login attempt, the IdPs jointly verify the user's password against the account information. The DPV process is critical for both DTG and DCI, therefore it is necessary for both IdPs and user to know if the process was successful in order to continue.

- **Generation of authentication material**: Once the users have been authenticated through the DPV system and depending on the scenario, they can choose between distributed token-based authentication or obtaining a privacy-enabled credential.

  - **Distributed Token Generation (DTG)**: The issuing of the token involves all partial IdPs for its correct generation and as a result, the user gets a number of fragments (token shares) that she will be able to combine into a short-lived full access token that includes only the minimum required data for the specific service.
  - **Distributed Credential Issuance (DCI)**: The issuing of the credential involves all partial IdPs so that the user obtains a number of credential shares, similar to what happens in DTG process. In this case, by combining these shares, the user obtains a full credential containing all the available attributes in the vIdP. The lifetime of the credential is longer than the DTG token, and it can be stored for later use. The user is able to generate presentation tokens with the minimum information required to access the desired services.

The typical flow of operation can be seen in the figure 3.4. We assume that the user is already registered in the IdM. The user selects a service to access (step 1). The SP communicates the access policy (step 2) and is redirects the client to the authentication portal (step 3). The authentication against the vIdP takes place via the DPV process and, once authenticated, the user is ready to receive the corresponding cryptographic material. The most common case is an online or SSO scenario where the vIdP will generate a presentation token via the DTG mechanism (step 4(a)). This mechanism implies that all partial IdPs forming the vIdP will issue a token share for the given access policy (if the user can meet the requisites of the policy). The user client is able to aggregate all the token shares and compose a valid presentation token (step 5(a)), with the minimum necessary data, for the access policy required by the SP. Once the presentation token has been reconstructed, it is sent to the SP who will proceed to its validation thanks to its DIP module (steps 6(a) and 7). Alternatively, a P-ABC can be obtained through the DCI mechanism (step 4(b)). In this case, all the partial IdPs generates credential shares that once reconstructed by the user client will result in a full credential with all the attributes that the vIdP has about the user (step 5(b)). The client's DCI module will be in charge of storing this credential securely and of directly deriving presentation tokens for a specific access policy (as long as it can satisfy it), with the minimum necessary information (step 6(b)). Once the presentation token is

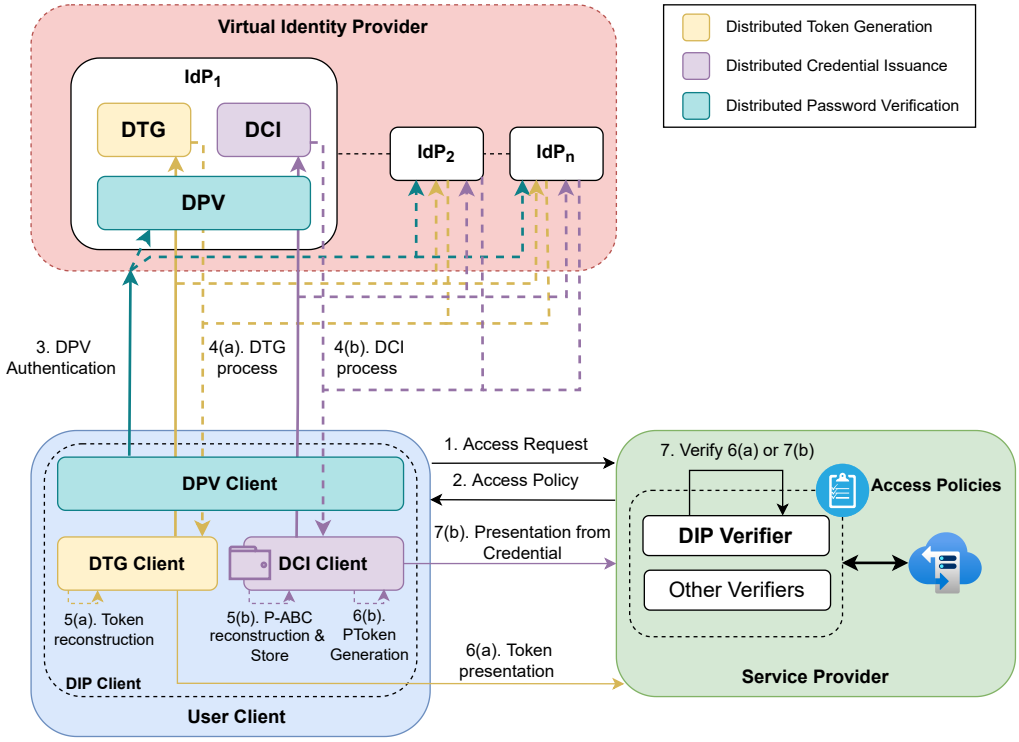


Figure 3.4: Distributed identity management system

generated, the client sends it to the SP (step 7(b)) who, is able to verify its validity (step 7).

3.4.3. Process definition

Distributed password verification (DPV)

The DPV process is based in the usage of pseudorandom functions (PRF) and particularly in partially-oblivious PRF (pOPRF) [79]. A PRF function takes as input a key of λ bits, an arbitrary message x and outputs a random looking string y. A distributed partially-oblivious PRF (dpOPRF), is a special type of multiparty PRF, where the key k is shared among n parties. Furthermore, the parties holding key shares, denoted $k_1, ..., k_n$, are not allowed to learn anything about the message x being queried. Finally, besides the actual message it is required that there is some tag, t, which is known to the parties holding the key shares. This dpOPRF operation implies that a message is firstly blinded by the querying third party, in order to hide its true value from the parties holding the key shares.

A dpOPRF system can be described by five processes:

1. **Setup**: Generates the public parameters pp for n parties for a specific instance, based on the security parameter λ.

2. **Key generation**: Takes the public parameters pp and generates a key share for each n participating entity.

3. **Blind**: Allows blinding an input value through an x-value that is indistinguishable from random. It also produces a value r that allows the unblind process.

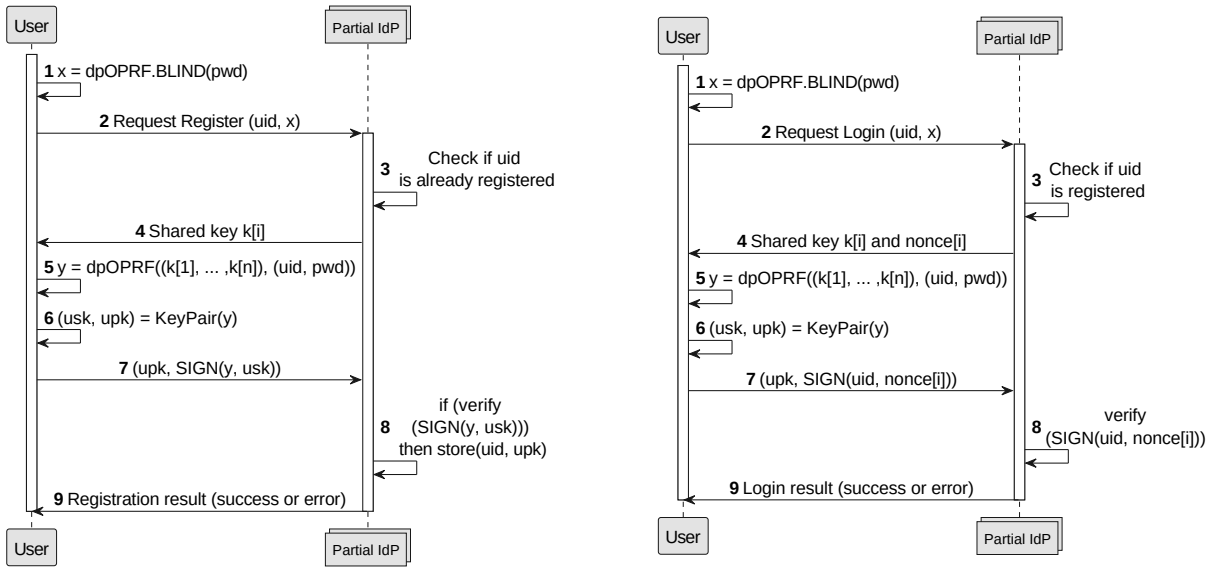
4. **Evaluation**: It takes as input a key k_i, a blind x value and a public label t and generates a PRF output y.

5. **Combination**: Combines the partial PRF outputs from Evaluation function, evaluates the blinded x and the label t and perform the unblind process to return the PRF output.

With this in mind, the DPV process is summarised in two stages: Registration and verification. In the registration phase a user requests to use the system by providing its username and password. The outcome is the user's account information that is stored at each of the single partial IdPs. During the verification phase, the user provides its username, uid and password to the partial IdPs. The outcome is an "accept" or "deny" from each IdP, indicating whether the user provided the password that corresponds to uid from the registration phase.

Registration phase requires the user to be assigned a unique identifier uid, together with the password pwd. Figure 3.5a shows the registration flow. Each partial IdP have a shared key k_i. First, the user performs a blind signature over the password and ask for a register process to the partial IdPs. The IdPs evaluate the registration request by looking if the uid already exists. If not, the IdP sends a response to the User containing the shared key k_i. The user now computes $y = dpOPRF((k_1, ..., k_n), (uid, pwd))$ using the result of the dpOPRF as a seed to generate a public and private key pair upk and usk. Then the user sends the generated public key upk together with a signature made with the corresponding private key, $(upk, SIGN(y, usk))$. Each partial IdP verifies the signature and stores the pair uid, upk as the user's access information. Finally, they send a confirmation message to the user.

Verification phase is initiated by the user, who sends a login request with the uid and the blinded pwd to the partial IdPs, figure 3.5b. The user holds uid and pwd and the IdPs hold the same k_i as in the registration phase, as well as optional account information (uid, upk) (in case the uid was already registered). Each partial IdP sends fresh nonce n_i in response to the user login request. Then, the user evaluates de dpOPRF operation as already described in the registration phase and derives a new keypair usk' and upk'. The user signs the uid and all the received nonces, using the upk'. The partial IdPs verify the signature against the stored upk and uid. If the verification passes, the login is successful.



(a) DPV Registration flow

(b) DPV Verification flow

Figure 3.5: Distributed password verification process

Distributed token generation (DTG)

The DTG process is based on the use of distributed signature (DSIG) [80–82] cryptography to construct a presentation token during the online or SSO scenario. A distributed signature scheme is a digital signature scheme, where shares of the private signing key are distributed between several parties such that the parties must collaborate in order to construct a valid signature. This scheme provides two very important features: key protection because all participants must be compromised in order to break the security and protection against forged signatures as long as all the participants must be malicious to generate a fake signature.

The DSIG scheme consists in five processes:

1. **Setup**: Given a security parameter λ and the number of participants n, generates the public parameters, pp.

2. **Key generation**: Given the public parameters pp, generates a public key vk along with a private key share sk_n for each participant.

3. **Sign**: Given a sk_n and a message m, output a partial signature share σ_n.

4. **Combination**: Takes a list of partial signatures shares $\{\sigma_1 ... \sigma_n\}$ and composes the full signature σ.

5. **Verification**: Given a public key pk, a message m and the combined signature σ checks if it is valid or not.

Figure 3.6 shows the DTG process based on DSIG. The vIdP is configured so that each partial IdP has its own sk_i ready for the DSIG.sign process. A user tries to access a service performing an access request and receiving an *AuthnRequest*. The authentication process is then carried out at the vIdP via DPV. If the authentication is successful, the user is required to provide the username along with *AuthnRequest*. The user chooses a random value *rand* and performs a $Hash(rand, req)$. The user sends to each partial IdP a message *msg* consisting of the hash and the *uid* to each of the partial IdPs, which will produce a token share. Once all the shares are received, the user reconstructs the token t. Finally, the user presents the t in the SP who can validate it using the DSIG.Verify operation using the vIdP public key pk.

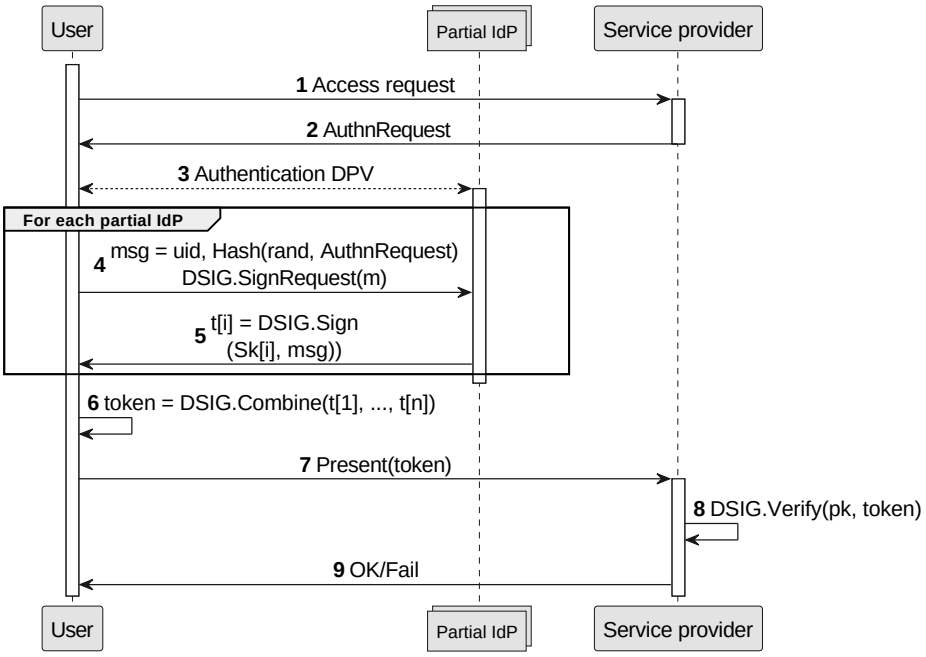


Figure 3.6: Distributed Token Generation (DTG) based on Distributed Signature (DSIG)

Distributed credential issuance (DCI)

The DCI process introduces a way to obtain privacy attribute-based credentials (P-ABC) in a distributed manner. The idea of P-ABCs was originally proposed by Chaum [83, 84], and several efficient realizations exist to date. The most prominent examples are the strong-RSA based and pairing-based schemes by Camenisch and Lysyanskaya (CL) [12, 77] the pairing based schemes by Boneh et al (BBS) [85], and Pointcheval and Sanders (PS) [86].

P-ABC solutions rely on an issuer's signature on a set of user attributes – the credential. From the credential the user can derive so-called presentation tokens that selectively disclose a sub-list of attributes. Given the revealed attributes, a verifier is guaranteed that a valid token was computed with a credential obtained from the issuer

for a super-list of those attributes. Importantly, the token reveals no more information about the user than what can be inferred from the revealed attributes. Originally, P-ABCs rely on a single issuer that provides users with their attributes, and thus poses a single point of failure. To remedy this recent works have shown how this can be achieved for pairing-based CL and BBS credential [80,87] while preserving the format of the resulting credential, which means that the user's derivation of presentation tokens is not impacted by this distributed issuance.

To integrate the credential issuing mechanism, a straightforward approach is followed which consists of directly modifying the DTG protocol to be able to issue P-ABC credentials instead of distributed signatures. That is, the user no longer receives signature shares from the IdPs, but rather shares of a credential containing all her user attributes. We then leverage the power of P-ABCs to let the user derive the final SSO token as a P-ABC presentation token from the freshly received credential, inheriting the unlinkability and minimal disclosure features from P-ABCs.

Leveraging on existing literature, the solution proposes the use of P-ABC credentials based on the scheme provided by Pointcheval and Sanders (PS) [86] as they have a good efficiency characteristics. However, no implementation of distributed version for these PS signatures was known.

Standard PS Signatures proposes an efficient signature scheme that allows to sign multiple message blocks $m_1, ..., m_k$ at once and also to efficiently prove knowledge of signatures in zero-knowledge proofs. In a similar way to the DSIG process, it has five basic operations:

1. **Setup**: Given a security parameter λ it generates the public parameters pp.
2. **Key generation**: Given the public parameters pp, generates a public key vk and a private key sk.
3. **Sign**: Given a message block $\{m_1, ..., m_n\}$ and the sk, produces signature σ.
4. **Verification**: Given a public key vk, a message block $\{m_1, ..., m_n\}$ and the signature σ checks if it is valid or not.

The next step is to evolve the standard PS Signatures into a multi-signature (MS) [88] model that allows a number $N \geq 1$ signatories to jointly compose a signature on a message, as we have seen in DSIG. Multi-signatures can be seen as a particular type of distributed signatures that allows for a more flexible key generation. Whereas a DSIG scheme generates a fixed set of signing keys at the beginning – often done through a trusted dealer, MS allow all signers to generate their independent signing key pairs and derive a joint aggregated verification key for any subset of signers.

The result of the combination of PS signatures with MS is detailed in the publication **(P6)**. The PS-MS signature scheme [89] used by DCI process is defined through six basic operations:

1. **Setup**: Given a security parameter λ, the amount of signers N and the number of messages to sign k, it generates the public parameters pp.

2. **Key generation**: Given the public parameters pp, generates a public key vk and a private key sk.

3. **Key aggregation**: Given a set of public keys $\{vk_1, ..., vk_n\}$ compute an aggregated public key $aggr_{vk}$.

4. **Sign**: Given a message block $m = \{m_1, ..., m_k\}$ and a private key sk_n computes a signature σ_N.

5. **Combination**: Given a set of σ_N, all the partial keys $\{vk_1, ..., vk_n\}$ and a message block $m = \{m_1, ..., m_k\}$ produces σ.

6. **Verification**: Given a signature σ, the $aggr_{vk}$ and message block $m = \{m_1, ..., m_k\}$, checks their validity.

With PS-MS it is possible to advance the implementation of a PS-MS based P-ABC system. The PS capability of creating efficient proofs of knowledge, immediately enables a (basic) P-ABC scheme [86]. Below we introduce an overview of the main processes resulting of mapping this P-ABC system to the PS-MS methods and the basic flow (figure 3.7). Full cryptographic details can be found at García-Rodríguez et. al **(P6)**.

1. **Key generation**: The N issuers generate their key pairs (sk_i, vk_i). Each issuer runs its own PS-MS setup and PS-MS key generation. PS-MS setup makes use of a k value to indicate the number of messages to sign and in this case, determines the number of attributes to be encrypted in the credential. The aggregate public key avk of the N partial IdPs is generated by PS-MS key aggregation such that $avk \leftarrow KeyAggregation(vk_1, ..., vk_i)$.

2. **Issuance**: When a user request a credential, the issuance process must generate a valid credential for a user uid with a list of attributes $A = \{a_1, ..., a_n\}$ valid for a time $epoch$. The N issuers run the signature method like $\sigma_i \leftarrow Sign(sk_i, (A, epoch))$ returning a signature share σ_i to the user. With all the shares, the user runs the combination method resulting in a full credential σ.

3. **Presentation**: To compute a presentation for a message m with a set of attributes $\vec{A} = \{A_1, ..., A_i\}$, the user must have the full credential σ. The user performs a knowledge proof such that: $token \leftarrow ZKProve(avk, \sigma, \vec{A}, R \subseteq [k], m)$.

4. **Verification**: Verifying a presentation token $token$ on a message m, revealed attributes $\vec{A}$ and an $epoch$ consists of verifying the signature of knowledge generated on m for an avk, as well as verifying that the $epoch$ has not expired, $result = ZKVerify(avk, token, \vec{A}, m)$.

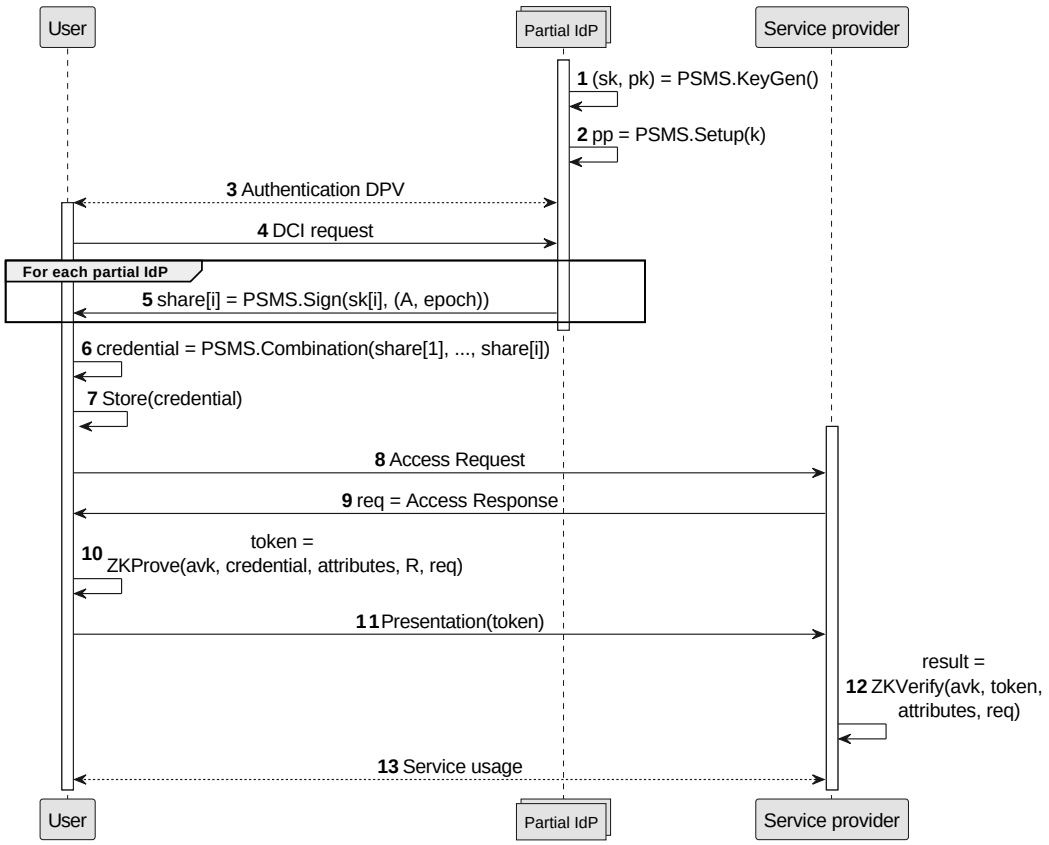


Figure 3.7: Distributed credential process flow

Initially, partial IdPs must generate the configuration material that includes the public and private keys as well as the number of attributes k that a credential will have (steps 1 and 2). The process starts with the authentication of the user through the DPV system (step 3) and then requests the issuance of a P-ABC credential via DCI (step 4). Each partial IdP generates a credential $share_i$ which is forwarded to the user (step 5). When the user has received all shares, it builds the full credential (step 5) and stores it for later use (step 6).

At a certain point, the user decides to try to access an SP so it initiates the connection with an access request and by receiving the access policy in req (steps 8 and 9). Using the stored credential, the user generates a presentation token and sends it to the SP (steps 10 and 11). Finally, the SP validates the received presentation token (step 12) and, depending on the verification result, provide the service (step 13).

Both distributed token generation (DTG) and distributed credential issuance (DCI) involve decentralized processes that enhance security, but they differ significantly in interaction requirements and use cases. DTG based on distributed-PABCs with PS-MS is non-interactive, allowing users to prove their credentials to the service provider (SP) without ongoing intervention from the identity provider (IdP), which is ideal for scenarios where constant connectivity is a challenge. However, DTG typically

requires the IdP to be involved continuously to validate and refresh tokens, making it suitable only for online environments where the IdP's presence is necessary for token validation. In contrast, DCI provides greater flexibility by allowing credentials to be issued once and then used repeatedly until they expire or are revoked, reducing the need for frequent interactions with the IdP and enabling both online and offline verification scenarios. DTG and DCI can be seen as complementary solutions in a comprehensive identity management framework. DTG, with its capacity for real-time authentication, is well-suited for environments requiring dynamic access control and immediate response to security events. Meanwhile, DCI's ability to provide long-lasting, reusable credentials makes it ideal for applications requiring durable identity verification, such as access to digital services or physical locations. By combining both approaches we can leverage the strengths of each to ensure robust security, flexibility, and user privacy across various contexts and requirements.

3.5. Conclusions

The solution presented in Chapter 3 emerges as a pioneering initiative aimed at addressing the perennial challenges surrounding digital identity management. Through its commitment to privacy, security, and decentralization, the solution seeks to transcend the limitations and risks endemic to centralized identity providers.

3.5.1. Primary Goals and Objectives

The solution envisions a paradigm shift in how personal data is handled during online interactions, centering on the following objectives:

Decentralization At its core, the presented solution advocates for a distributed approach to identity management. This strategy aims to dilute the reliance on singular, centralized entities, thus mitigating potential central points of failure and enhancing overall system resilience.

User Convenience Despite its rigorous security and privacy measures it prioritizes user experience, ensuring that these enhancements do not encumber user convenience or accessibility.

3.5.2. Technological Underpinnings

It introduces a sophisticated blend of advanced cryptographic techniques. These technologies are meticulously integrated to ensure scalability and interoperability, thereby paving the way for a comprehensive and adaptable identity management ecosystem.

3.5.3. Anticipated Impacts

Its implementation heralds a new era characterized by:

- Enhanced security and privacy, significantly reducing vulnerabilities inherent in centralized systems.
- Broadened adoption of privacy-centric services, spurred by heightened user trust and regulatory compliance.
- Stringent adherence to regulatory standards, including GDPR, demonstrating a commitment to user rights and data protection.

3.5.4. Encountered Challenges

Despite its promising outlook there are open challenges, including:

- Technical intricacies associated with the deployment and maintenance of distributed cryptographic systems.
- The steep learning curve and user education imperative for fostering trust and facilitating a smooth transition from legacy systems.
- Interoperability dilemmas, necessitating seamless integration with existing digital ecosystems without compromising security or user privacy.

3.5.5. Contemplated Drawbacks

In addition there are some potential drawbacks identified, notably:

1. Technical complexities that may introduce latency or reduce system efficiency, particularly in identity verification processes.
2. Challenges in user adoption, influenced by trust issues and the necessity of acclimating users to a novel system architecture.
3. Legal and regulatory hurdles, especially in navigating the intricate landscape of global data protection laws.
4. Economic and social ramifications, including the risk of widening the digital divide and the resource-intensive nature of deploying and sustaining the proposed framework.

Final Thoughts In essence, Chapter 3 represents a forward-thinking endeavor that seeks to revolutionize the domain of digital identity management through a meticulous blend of decentralization, privacy, and security. While the path to its full realization is fraught with challenges and considerations, the potential benefits for users, service providers, and the broader digital ecosystem are undeniable. As we look towards the future, the solution has been integrated as part of the OLYMPUS project and it success will hinge on our collective ability to navigate these complexities, ensuring its widespread adoption and operational efficacy.

CHAPTER 4

DLT-enabled identity management system with enhanced trust

4.1. Introduction

Chapter 3 introduces a new proposal for identity management applying distributed techniques, intending to solve the problems presented in traditional solutions and even in new distributed systems. The approach devises a privacy-preserving identity management solution evolving from federated identity systems and eliminating the IdP as the single point of failure. The above proposal identifies a number of challenges in Section 3.3. Achieving these objectives makes the above proposal a more private, usable and secure solution. However, there is still room for improvement.

Despite the advantages over traditional systems, the architecture 3.2 can be improved in terms of trust. Although chapter 3 introduces distributed technologies through the distribution of the monolithic IdP into several partial IdPs, and the issuance of cryptographic material, it does not address the trust relationships that are still necessary between the entities involved. The decomposition of the IdP into several partials does, to some extent, manage the trust between the partial IdPs themselves but neglects the trust relationships between users, vIdP, and service providers which they still need to trust, as traditionally done, blindly.

There are three key elements of trust, namely: the user trusts the identity provider, the service provider trusts the identity provider and the user trusts the service provider. Traditionally, a corrupt identity provider can potentially access the service, using the user's identity without consent (impersonation), a situation that is solved in the above

distributed approach. For its part, the service provider must ensure that the identity provider is trustworthy since it will delegate to the identity provider the collection and validation of user attributes. Finally, the user must be able to trust that a service provider is acting legitimately and transparently before sharing his or her information. Too often, service providers misrepresent their requirements or omit the amount of data they collect and by the time users become aware of the amount of data being shared, it is often too late.

From the perspective of the trust transfer between SP and IdP in most deployments this will not be problematic, as the IdP will be deployed by some trusted organisation. In some cases, however, the IdP acts as some kind of proxy. This situation requires that the service provider trusts not only the IdP itself but the organisation deploying it.

Regarding the managing of trust and key material typically, service providers trust an identity provider by installing the IdP's crypto material in some trust store. This can be somewhat inconvenient for the service provider and instead common PKI technology is often used, by having some certificate authority (CA) sign the IdP's public key in the format of an X.509 certificate. This essentially binds a domain name to a public key, allowing the service provider to trust the tokens signed by some X.509 certificate. The above solution supports this type of key distribution but, to prevent a vIdP to link a user to a service provider, the key material should be exchanged out of band. In addition, although the service provider can obtain the key material in multiple ways, it must also obtain a description of the vIdP. Traditional solutions such as OIDC or SAML achieve this by using IdP metadata messages. These metadata description messages contains relevant information such as various URLs, endpoints and information regarding the key material. The same approach can be taken for a vIdP, although we can no longer make use of the standard messages as a vIdP is not exactly a traditional IdP as it consists of multiple servers, endpoints etc. Existing metadata specifications would have to be extended to support the vIdP key material distribution.

It should also be considered that users are reticent to change any authentication solution unless the benefits are obvious and do not force them to change their behaviour. Generally, the trust relationships that are established in IdM systems require the user to simply trust without providing any extra information. For full trust to really exist, the system needs to provide more information or tools to resolve the situation without detracting from the user and service providers' experience.

To enhance the strengths of the distributed IdM system 3, it is necessary to introduce features to successfully manage trust between the different entities. To this end, we take advantage of Distributed Ledger Technologies (DLT) [5] to enrich the solution. The popularisation of these technologies and the identity solutions shown in 2.2 and 2.2.1 are empowering users with better security and privacy mechanisms [8] that allow them to become anonymous and re-take control of their data with self-sovereign identity (SSI) models.

4.2. Concept

The proposed evolution aims to substantially improve trustness in the entire infrastructure without penalizing the user experience and maintaining the precepts of ease of use, deployment, and integration with other technologies.

Chapter 3 offers a distributed identity management solution with two modes of operation, (1) online and (2) offline. After analysing the architecture and the necessary trust relationships, there are several areas where improvements can be made. Firstly, from the perspective of the trust transfer between SP-IdP and user-IdP, although the composition of the vIdP is supported by cryptography, the distribution of public material such as vIdP components, their keys and url endpoints etc., must be done either by manual configuration, or through the extension of traditional SAML or OIDC messages. This situation is not at all convenient for any of the entities. Moreover, users and SPs must trust the legitimacy of the vIdP (including partial IdPs) with no extra information as if it were a traditional system.

Secondly, the trust relation between users and SPs remains as always. Users must take service access decisions according to a set of policies with no other help than their common sense or through the false security provided by seeing a small padlock on a website indicating that a TLS connection exists. Experience has shown that relying solely on the common sense of users often leads to problems of personal data leakage or worse, phishing, and even loss of credentials and personal data. Even though users have learned the importance of a TLS-protected connection, they have not fully understood that a TLS enabled site can also be fraudulent or dangerous. Users tend to trust that everything will work as expected as long as they do not have clear feedback that something might be wrong.

We know that DLT systems can operate in parallel to virtually any infrastructure, providing confidence through features such as immutability. The approach seeks to evolve distributed system towards a DLT supported distributed system, figure 4.1. The objective is to create a complete ecosystem where, in addition to protecting user privacy, trust management between the different entities is enhanced through the DLT, that acts as a source of trust for all participants enabling the obtention of relevant information about the vIdP, public cryptographic material and about the service providers operating in the framework.

Integration with DLT technologies is sometimes too dense or complex for successful implementation. The proposal should maintain the ease of use and transparency already achieved in the previous proposal. There would be no point in improving trust mechanisms if the solution becomes unusable. The new approach should emphasises in the registration and discovery through DLT of the vIdP and service providers. To this end, technologies such as smart contracts provide the necessary functionality. In any identity management scenario there are two basics steps: (1) The setup, in which the user must select the IdP and either manually configure some kind of parameter or rely on those that the corresponding application downloads from an external source or from the IdP itself and (2) the usage, where the user sign up with a set of attributes

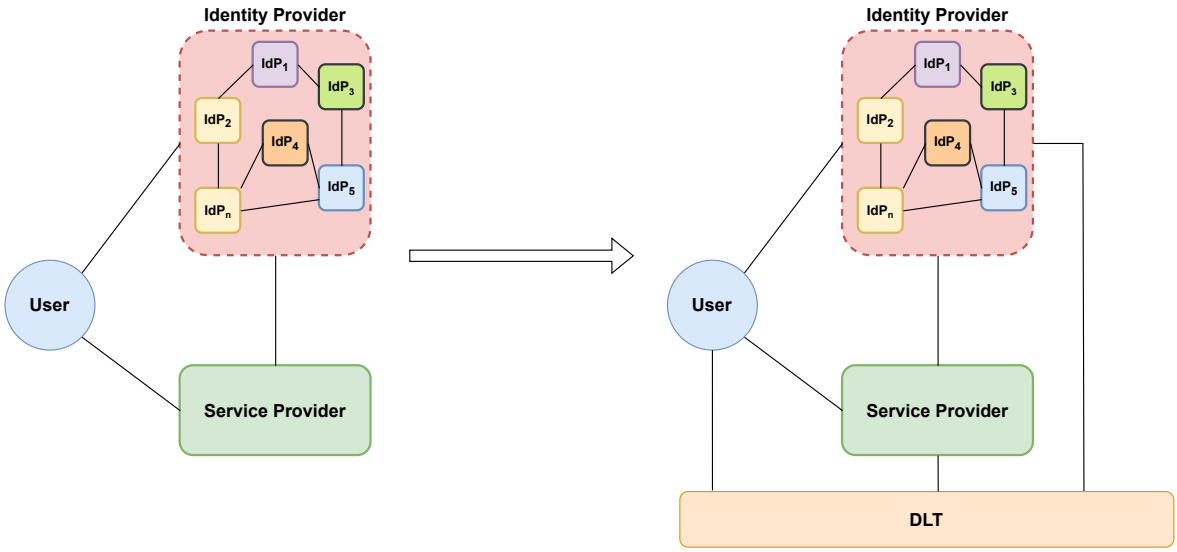


Figure 4.1: Conceptual idea

to obtain some material to later present in a service provider. In both cases, there is no guarantee that there could have been any malicious modification of the vIdp or SP parameters. Similarly, when a user wants to make use of a service that is available through the identity provider, the user is not able to know a priori whether that service provider is more or less trustworthy. It is only able to discern whether the connection it offers is secure or not, which leaves the user at a disadvantage in the face of a possible threat. The proposal tries to provide a solution to these scenarios of loss or absence of trust.

4.3. Objectives and requirements

Chapter 1 presents a set of general objectives and requirements together with a set of specific objectives (section 1.3). This chapter encompasses objectives **O5** and **O6**, building on the results obtained in chapter 3 and leading the proposal to the following key areas points:

- Novel identity management solutions with DLT technologies still rely on centralised identity providers.
- Existing solutions, with and without DLT, lack trust management.
- The introduction of DLT technologies has only led to an improvement in traceability. Trust is slightly improved in a collateral way.
- Existing solutions with DLT technologies remain unwieldy for users.

Having identified those gaps, **O5** focuses on developing an advanced IdM DLT enabled solution based on the solution obtained in chapter 3 that should address the following challenges:

Challenge 1 Provide a distributed IdM system along with DLT technologies.

Challenge 2 Maintain the security an privacy capabilities without penalising the user experience.

Challenge 3 Prevent the IdP from tracking or impersonating its users.

Challenge 4 Reduce or eliminate potentially dangerous situations during access to the IdP.

Challenge 5 Reduce or eliminate potentially dangerous situations during service access.

Challenge 6 Provide user-friendly tools to improve decision capacity and control over private data.

Challenge 7 Keep minimum requirements low in order to maximise the number of devices that can support the solution.

In addition to the security and usability requirements set out in the section 3.3, tables 3.1 and 3.2, the following requirements have now been added regarding the ledger:

RQ.ID	**Name**	**Description**
ledger.RQ.1	Ledger data controller	The ledger stores public information and under no circumstances does the ledger store private data.
ledger.RQ.2	Ledger deployment	Setup of a public, permissioned and lightweight ledger to securely store and share trust information and access policies.
ledger.RQ.3	Ledger read and write	Implementation of smart contracts to carry out the interaction between the different entities and the DLT infrastructure.
ledger.RQ.4	Ledger verification	All information contained in the ledger must be verifiable and traceable.
ledger.RQ.5	Ledger and solution evolution	The DLT platform should not be coupled with the identity solution. In other words, the ledger solution is independent from the IdM.
ledger.RQ.6	Ledger usage	The ledger acts transparently and does not penalise or restrict the operations.

Table 4.1: Ledger requirements

In the following sections we will discuss the basic building blocks applied to the solution as well as the processes and the proposed architecture. This will be followed by an evaluation through a set of use cases and finally the main results obtained.

4.4. Processes and architecture

4.4.1. Overview

In the proposal made in Chapter 3, the main objective of the processes and entities was to eliminate the traditional IdP as a single point of failure and to achieve an identity system based on distributed technologies. Having achieved that goal, the challenge now is to improve the trustworthiness of the entire infrastructure.

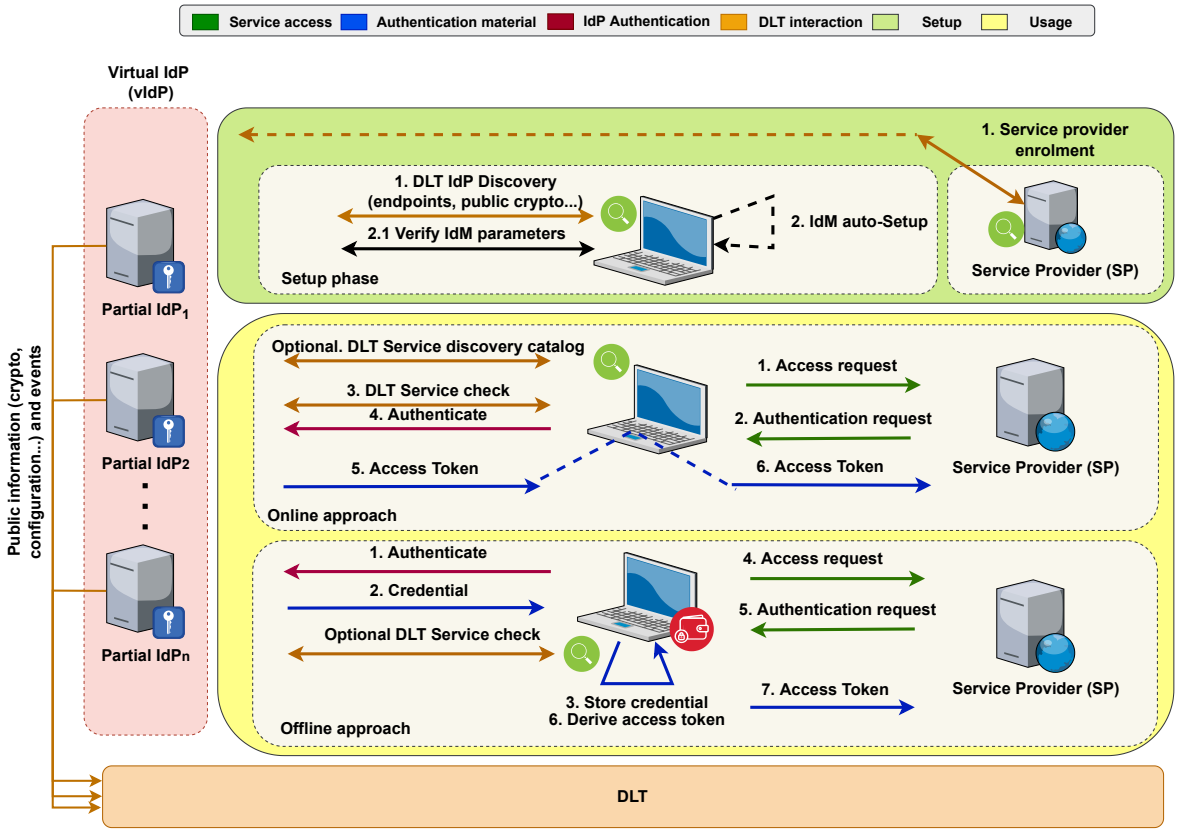


Figure 4.2: DLT enabled IdM evolution proposal

This approach 4.2 distinguish two basic phases. On the one hand, the (1) **configuration or registration phase**, figure 4.3 and on the other hand, (2) **the identity management phase**, figure 4.4.

(1) Registration phase This phase takes place during the deployment of new identity providers or new services. During it, the objective is to get the vIdP and the SPs with

which the users will subsequently operate registered on the DLT platform. While the registration of a service provider is not complex, in the case of vIdPs it is a critical process. The vIdPs are virtual entities composed of several partial IdPs that may be distributed over different domains. Therefore, it is necessary to register separately all partial identity providers that form a vIdP.

(2) Identity management phase The system is ready to be used by different users who are able to perform authentication and authorization operations in addition to query relevant information about vIdPs and SPs directly through the DLT platform. This information includes cryptographic parameters, endpoints and policies of the different registered vIdP and service providers.

4.4.2. Architecture

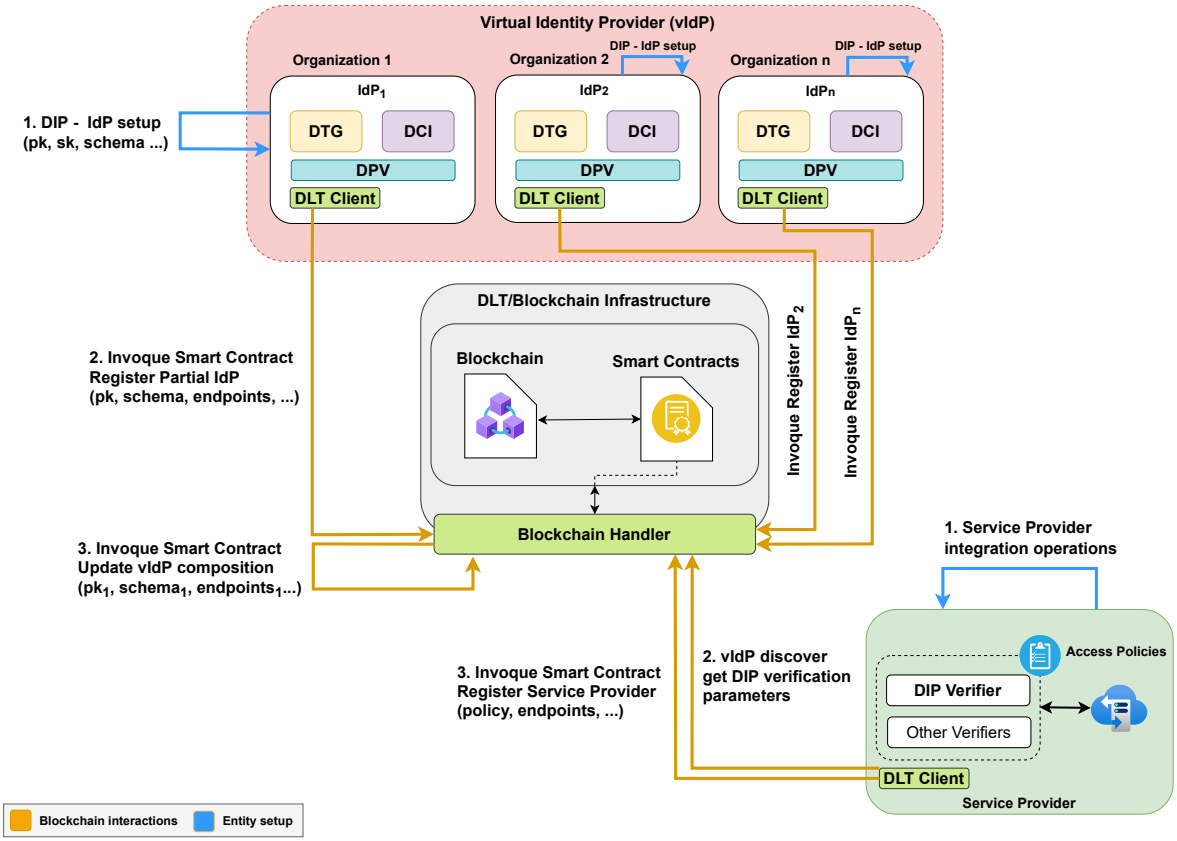


Figure 4.3: DLT enabled IdM evolution, Phase 1 - Registration

The figures 4.3 and 4.4 show a detailed view of the two phases configuration and IdM respectively of the architecture, the entities involved and the relationships between them. With respect to the previous solution, the introduction of the DLT entity is striking. This architecture proposes the inclusion of Blockchain technology with support for smart

contracts. The blockchain infrastructure is a public-permissioned platform which its operation is intended to be more ambitious than a simple distributed database, it acts as a source of trust distributing public cryptographic material, connection information or even reputation about the different entities. To achieve such integration and since every Blockchain platform has its own set of operations, APIs etc., the approach introduces an adaptative interface called **Blockchain Handler**. The purpose of this element is to homogenise the access to whatever DLT solution running in the background, avoiding extra complexity when changing the system to another DLT solution or updating the to newer versions.

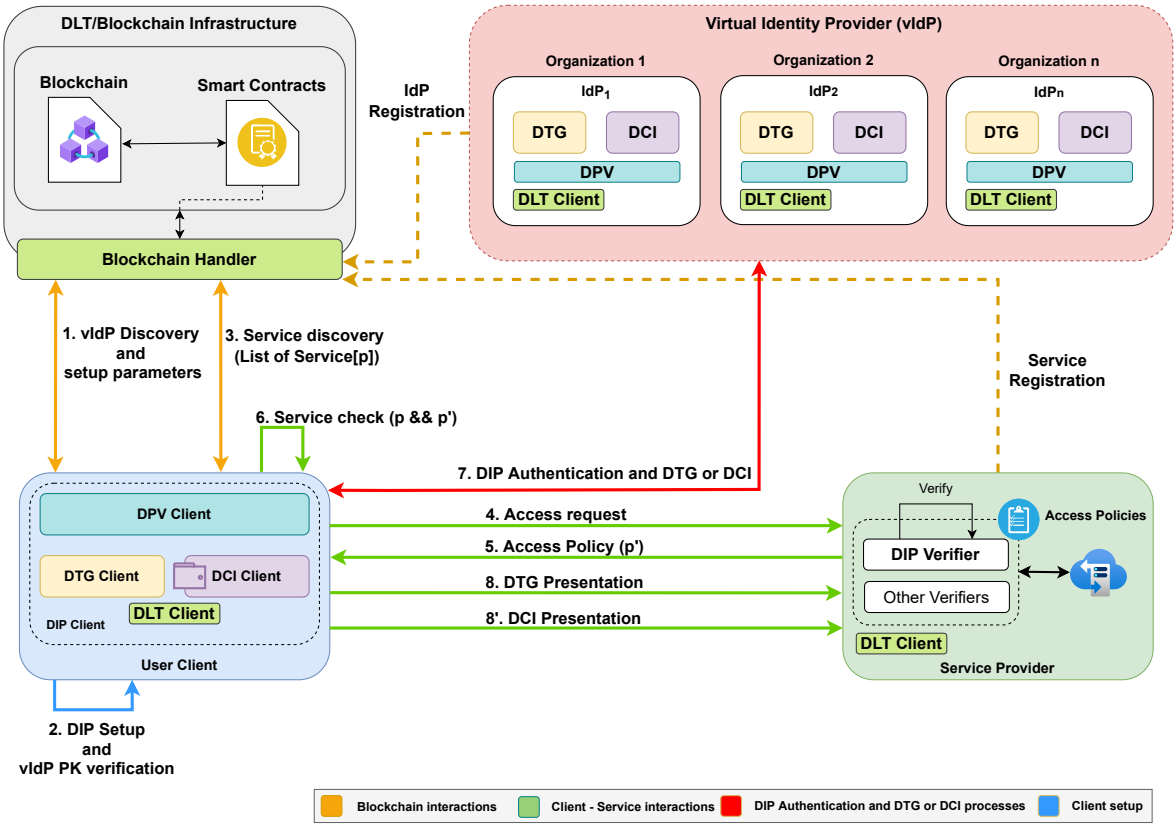


Figure 4.4: DLT enabled IdM evolution, Phase 2 - Identity Management

The previous solution assumed an ideal scenario where the vIdP was comprised of different partial IdPs located in the same domain (e.g. the same company). The solution did not include any mechanism for the transfer of trust between domains. Now, it is no longer assumed that the vIdP is deployed in a single domain but may actually be distributed across different ones using the DLT infrastructure as a secure, tamper-proof element for transferring and querying relevant data. Partial IdPs include, in addition to the DIP-related modules, a DLT module. The **DLT Client** is in charge of bringing connectivity to the existing DLT deployment through the **Blockchain Handler** and it is designed to be replaceable and/or extendable so that the solution is not tied to a

single DLT deployment. The same module appears in the user and SP sides, providing DLT connectivity.

Although the proposal does not modify the underlying cryptographic processes DPV 3.5, DTG 3.6 and DCI 3.7, it does sightly alter the behaviour of the different entities. The **user** actively interacts with all entities. Starting with the DLT, from where the user obtains a catalogue of trusted identity providers together with their configuration parameters allowing the user's client to perform a self-configuration process. In addition, it internally verifies that the information retrieved from the DLT actually matches with the chosen vIdP (i.e., by computing the aggregated public key). After client configuration, the client retrieves a list of the registered services including the information about the consumed data or access policies. Once the service is selected, the user client is able to crosscheck the access the real service information against the DLT stored one. If something has changed, the user is informed so that he or she is able to make a decision before sharing any personal data. The added DLT based verifications do not alter the authentication and the obtention of tokens or credentials. User actions are only required in case of potentially dangerous situations.

The behavior of the **service provider (SP)** is also modified. Now the service provider must be registered in the identity management platform, providing a set of basic data such as the private data it consumes, its endpoints etc. In this way, any service supported by the solution is also guaranteed by the DLT. Similarly, the solution requires the registration of the **Virtual IdP** in DLT. Given that the vIdP is a virtual entity, the registration requires that each of the partial IdPs be registered separately in the DLT. The registration includes all the basic data such as public keys, endpoints, etc., so that any entity querying the DLT is able to obtain an unmodified version of them and even cross-check it with the actual entity.

As a result of the adjustments made to the entities behaviour, it can be seen that the registration phase is very important, as the correct enrolment of the entities subsequently guarantees the correct functioning of the entire infrastructure. Assuming the modifications it is necessary to model the flows for the registration as well as the data models that will support the trust parameters. The vIdP together with the partial IdPs are the critical entities to be registered. As we know, the vIdP is composed of a set of partial IdPs that at the time of launching must perform an independent registration process against the DLT infrastructure so that each of them separately indicates its address, public cryptographic material and other relevant parameters. The registration of the vIdP, on the other hand, is more challenging. The vIdP cannot be registered like partial IdPs because it does not really exist. It is a logical construct that does not operate on its own. Nor can we delegate its registration to any of the partial IdPs that form it, since we would then be giving greater responsibility to one IdP than to the others, generating a point of failure and a possible attack vector. To ensure that the IdP and vIdP are correctly registered, the solution relies on the use of smart contracts installed on the DLT platform. Each time a new IdP is deployed, a specific smart contract must be invoked to register it in the infrastructure. In the same way, the registration of the vIdP will be given by a smart contract that will be automatically

launched from the DLT platform itself each time a new partial IdP registration (or update) is done, avoiding that any of the partial IdPs acts as controller.

Service providers must also be registered with the DLT in order to provide a fully reliable experience. Their registration is also done through a smart contract that collects and inserts into the DLT their connection parameters, description and the user's private data they expect to consume when providing the service.

While the advantages of integrating the Blockchain Handler are readily apparent, it is imperative to maintain a comprehensive awareness of the potential challenges that such incorporation may introduce. These challenges are critically summarized as follows:

1. **Complexity and Overhead**: The addition of the Blockchain Handler introduces an additional layer of complexity to the system, which may lead to increased developmental and operational overhead.

2. **Scalability**: The Blockchain Handler might constrain the system's ability to scale effectively, potentially limiting the utilization of specific features and optimizations inherent to various Distributed Ledger Technology platforms. Each blockchain platform has its unique set of features and optimizations, and a universal handler might not fully leverage these specific benefits, leading to potential bottlenecks.

3. **Dependency and Rigidity**: Reliance on the Blockchain Handler for interfacing with different blockchain platforms may induce a degree of inflexibility, complicating updates and adaptation to new technologies. This central component, crucial for the operation of DLT services across various entities (like user, SP, and IdPs), becomes a prime target for security attacks.

4. **Technology Lock-in**: Although designed to facilitate flexibility, the Blockchain Handler could inadvertently lead to a new form of technology lock-in, where the system becomes overly dependent on the handler's current capabilities and supported platforms.

It is essential to consider these factors carefully to ensure the robust and flexible deployment of the Blockchain Handler in the architecture. However, despite of these challenges the introduction of the Blockchain Handler in this solution might not pose significant problems and could indeed serve as a beneficial component. These advantages are outlined as follows:

1. **Unified Interface:** The Blockchain Handler provides a standardized interface for accessing various Distributed Ledger Technologies, simplifying integration and management across different blockchain solutions.

2. **Facilitates Upgrades and Maintenance:** It abstracts the complexities of individual DLT platforms, enhancing maintainability and ease of upgrades, thus minimizing system-wide disruptions.

3. **Reduces Developer Burden:** By offering a common set of APIs, the Blockchain Handler alleviates the need for developers to understand the intricacies of each blockchain platform.

4. **Enhanced Security and Compliance:** Centralizing blockchain interactions through a single component allows for uniform security measures and compliance standards, potentially increasing the system's security posture.

5. **Scalability through Abstraction:** The abstraction layer can manage and optimize requests to the blockchain, facilitating scalability even in complex operations.

6. **Support for Smart Contracts and Complex Operations:** The handler provides a manageable way of deploying and interacting with smart contracts, ensuring consistent and reliable execution.

7. **Reduction in Technology-specific Risks:** By decoupling core system operations from the specifics of the underlying DLTs, the handler reduces risks associated with technology-specific failures or limitations.

The incorporation of the Blockchain Handler into the DLT-enabled Identity Management (IdM) system strategically aligns its capabilities to meet the specific challenges of the architecture. Firstly, the Unified Interface and Scalability through Abstraction directly address the need for a distributed IdM system by facilitating seamless integration across various DLT platforms, ensuring the architecture's scalability and adaptability (**Challenge 1**). The Enhanced Security and Compliance aspect crucially maintains user privacy and security, and prevents identity providers from misusing user data (**Challenge 2** and **Challenge 3**), enhancing trust and reliability in the system.

Furthermore, the ability of the Blockchain Handler to streamline upgrades and maintenance not only preserves the system's security and user experience but also ensures continuous alignment with evolving technological standards (**Challenge 2**). The Reduces Developer Burden feature is particularly vital in maintaining minimal system requirements, thereby expanding the technology's accessibility and utility across diverse devices (**Challenge 7**).

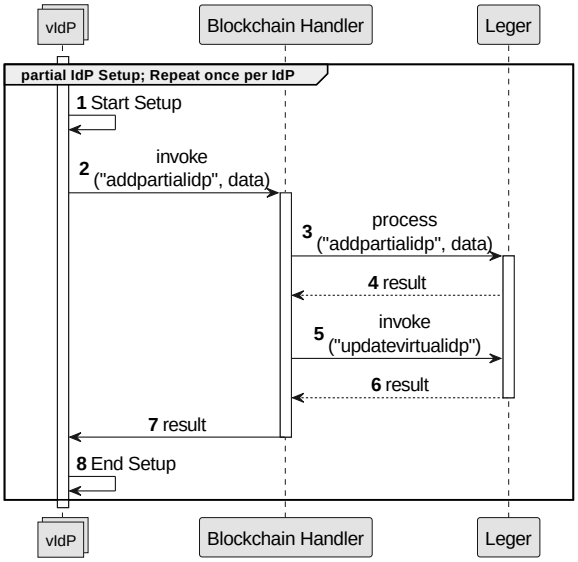
Additionally, the Support for Smart Contracts and Complex Operations offered by the Blockchain Handler mitigates risks during critical interactions with identity providers and service access, thus safeguarding against dangerous situations (**Challenge 4** and **Challenge 5**). Lastly, the Reduction in Technology-specific Risks promotes a user-friendly environment that empowers users with enhanced decision-making capabilities concerning their private data (**Challenge 6**). Altogether, the Blockchain Handler adeptly bridges the gap between advanced technological capabilities and user-centric security and operational needs, underpinning the robustness and efficacy of the IdM system.

The following subsection will delve deeper into the process definition. This part of the discussion will outline the specific procedures, protocols, and workflows that are

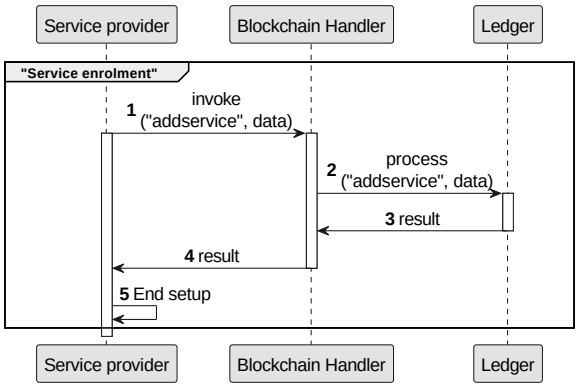
fundamental to rendering the described advantages, ensuring they effectively address the identified challenges.

4.4.3. Process definition

The registration phase (1), figure 4.5, is divided into two separate processes. On the one hand, the registration of vIdPs and partial IdPs, figure 4.5a, and on the other hand, the registration of service providers, figure 4.5b.



(a) IdP and vIdP register



(b) Service provider register

Figure 4.5: Registration phase

Each partial IdP autonomously initiates its own configuration process, generating the corresponding cryptographic material (step 1). When the IdP has sufficient information available to register, it proceeds to invoke the registration smart contract *addpartialidp*, to which it provides the necessary data (step 2). The contract internally checks that the IdP that is trying to register does not previously exist in the DLT and if everything is correct (steps 3 and 4), it continues with the registration. In parallel, the smart contract itself invokes a subcontract that takes care of generating or updating the registration of the vIdP to which the IdP being registered is associated (steps 5 and 6). Once both processes are completed, the partial IdP receives a response with the status of its registration and the vIdP data (step 7) finishing the setup process (step 8). The SP registers in the system by means of the *addservice* contract and a series of descriptive data such endpoints or consumed user data (i.e., user email). The use of the blockchain entity handler makes the process as simple as HTTP(s) connection or similar, thus avoiding tedious protocols. We only need a secure connection based on web standards such us TLS to be able to register against the IdM system. This is very advantageous from the point of view of the users, who will not need powerful hardware,

but also for the SP administrators, who do not have to learn strange processes or use unfamiliar technologies to register their services.

Since the SP also acts as a verifier, it must perform an additional configuration process. Previously, this process was performed manually by the service administrators but now, by leveraging smart contracts, we are able to auto-provision the necessary information directly from the DLT while maintaining security and trust.

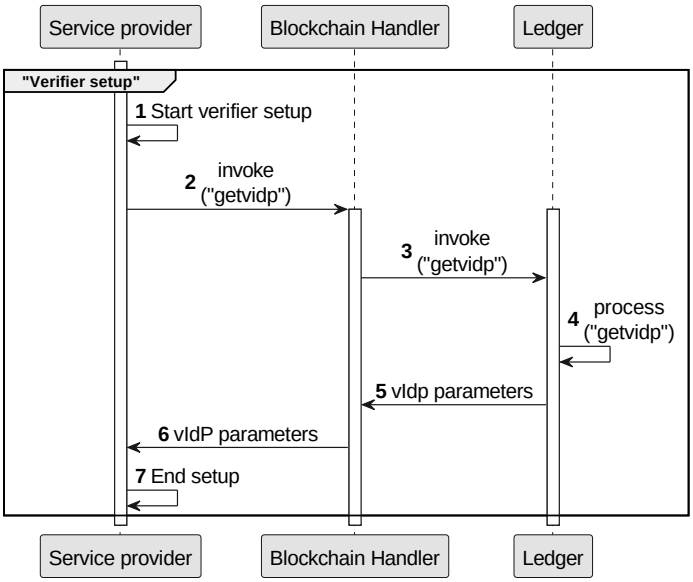


Figure 4.6: Service provider auto-setup

From the user's perspective, the operation of the solution also changes slightly from what was introduced in the previous chapter. The user is the central element to be protected. In the previous approach, user protection is mainly enhanced through a new identity provider model, which bases its operation on distributed cryptography, avoiding typical problems such as tracking or impersonation. In addition, it also benefits from the principle of minimal-disclosure, avoiding having to give more information than necessary. This approach extends the protection of the user through the enhancement of trust. The user used to rely on the honesty of all the elements. The user chooses an identity provider, without having any additional information in the same way as she chooses to make use of an SP, and chooses whether or not to give her data once she has already made the first access attempt. This approach eliminates the need for blind trust and provides relevant information with which the user is able to make better decisions and protect her privacy. The configuration of the identity provider becomes automatic by leveraging the DLT support. The user's interaction is limited to selecting the most convenient for her, just as in traditional solutions she chooses between Google, OpenID, Facebook, etc. All configuration data comes in the first instance from a source that cannot be altered and is publicly auditable. Once the client is configured, the user can consult a list of available services together with a description showing the registered information. When the user decides to access a service, the information provided by

the service and the information stored in the DLT is silently checked to ensure that it matches. In case the silent check fails, the user receives an early warning that would allow her to abort the process before sharing any personal information.

The figure 4.7 shows the auto-configuration process of the client. It is carried out through the blockchain handler. The client asks the handler for the list of available vIdPs through a smart contract that retrieves this information from the DLT (steps 2 to 5). The user then selects the vIdP of interest and, through the handler and by invoking a query contract, obtains the concrete data for the selected vIdP (steps 6 to 9). With this data, the user client is able to learn the necessary information to communicate with the vIdP and start operating. The process is as simple as possible in order to make its integration as immediate as possible in any commonly used application. It would even be possible to skip the selection of vIdP and directly pre-configure a specific one so that the user does not even have to intervene.

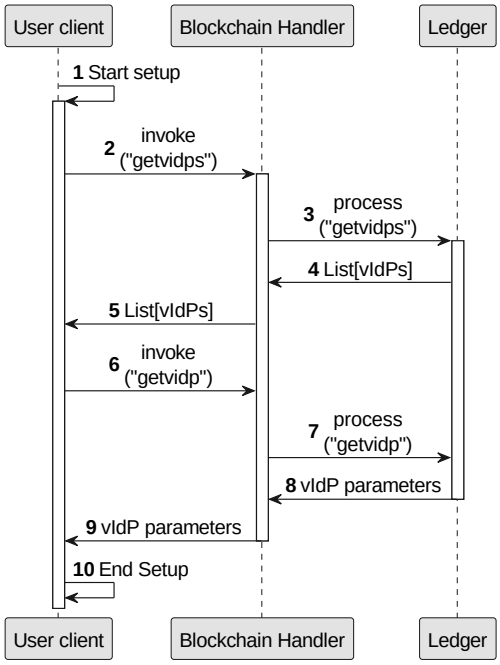


Figure 4.7: User client setup

The figure 4.8 shows an example of a complete interaction in which a user try to use a service. This interaction includes the IdM stages summarized for convenience. First, the user retrieves the existing services from the DLT (steps 1 to 4) and selects one (step 5), receiving a set of parameters (i.e., access policy) (step 6). Next, the user client silently invokes the *getService* contract to retrieve the service record and the data associated to it (steps 7 to 9). The application compares the information received from the SP, s' with the s recorded in the ledger, warning the user if something has changed (step 11). At this point, the user visualizes the policy applied to access the service and makes a decision (step 12). If the user continues, the next step is to authenticate within

the IdM using the distributed identity protocol to perform the DPV 3.5 authentication and to obtain the authorization material via DTG 3.6 or DCI 3.7. Finally, depending on the method used, the user client combines and generates or forwards the presentation token to the service provider for verification and to provide or not the required service (steps 15, to 19).

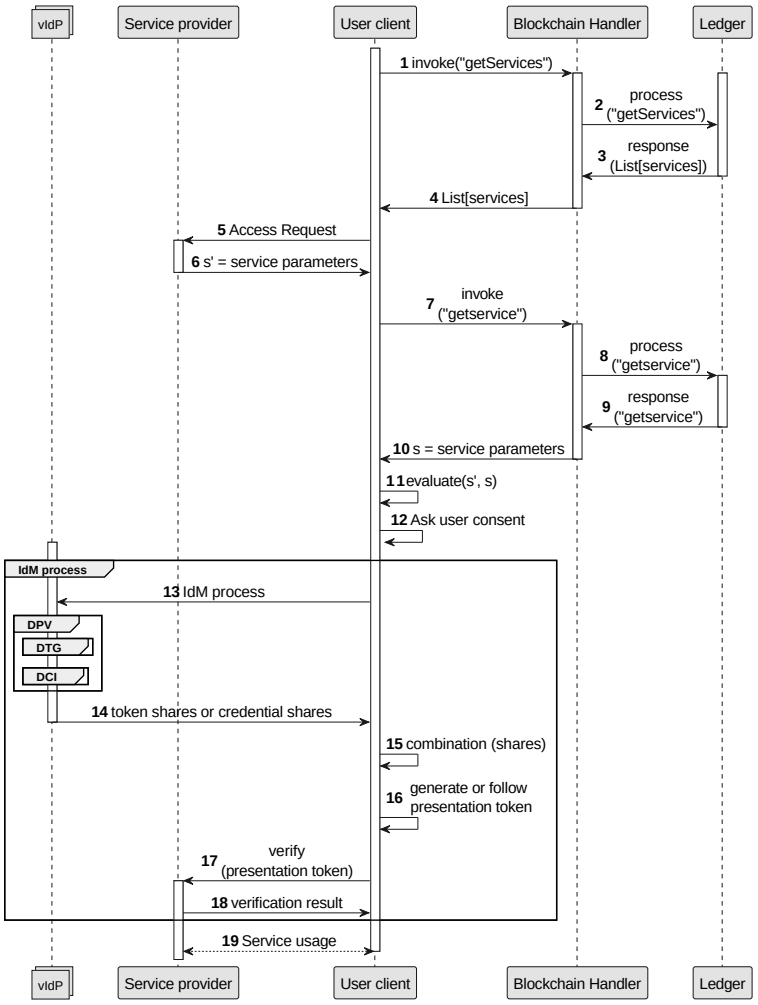


Figure 4.8: User client usage

The inclusion of DLT not only does not penalise the operation of the system, but also adds an extra layer of trust. Smart contracts enable the detailed and secure registration of critical infrastructure components such as vIdP, partial IdP and SPs, making the DLT infrastructure act as a verifiable source of trust. Moreover, users benefit from a practically automatic setup system for their identity provider and a new protection mechanism that allows them to silently assess whether the service they are trying to access is operating as expected. Furthermore, it is possible to query the public data of any of the entities registered in the DLT at any time, for example, by performing random checks simply to detect possible changes or to update access routes. This has

operational advantages, e.g. SP administrators can automate checks and update their services based on data received from the DLT.

To facilitate the implementation of the newly introduced processes and the operation, it is imperative to delineate a series of data models. These models will be instrumental during the entity configuration phase and subsequently in the issuance of the requisite cryptographic materials throughout the workflow. This structured approach ensures a systematic and secure integration of the components within the specified processes.

Initially, it is crucial to clearly define the entities involved, with specific focus on the partial identity providers and the virtual identity provider. These components play a vital role in the overall functionality and integration within the framework. Subsequently, it is equally important to equip the service providers with a well-defined structure that enables them to effectively operate within the proposed solution.

In the context of partial identity providers, they are delineated utilizing the data model presented in Listing 4.1. The JSON data model for a partial identity provider includes several key properties, each of which plays a crucial role in the identity management system:

- **status**: Indicates the operational status of the partial identity provider. Values such as "ACTIVE" or "INACTIVE" reflect the current state of the provider, affecting its availability for transaction processing and interaction.
- **publicKey**: Contains the cryptographic public key associated with the identity provider. This key is essential for ensuring secure communications and verifying the integrity of transactions through digital signatures.
- **spawnDate**: Records the date and time when the identity provider was established. This timestamp is critical for auditing, logging, and maintaining the security of the identity management system.
- **did**: This nested object represents the Decentralized Identifier (DID) associated with the identity provider and includes:
  - **id**: A unique identifier conforming to the DID specification, providing a global identifier that is unique within the system.
  - **context**: A URL pointing to a JSON-LD document that defines the schema used in the DID document, crucial for ensuring that the document's terms are interpreted consistently.
  - **service**: Details the services offered by the identity provider, encompassing:
    - **serviceEndpoint**: Specifies the URL or other service access points for interacting with the identity provider.
    - **type**: Describes the function of the service, in this case, "Partial-IdP", which identifies the role of the entity in the identity management ecosystem.

```
{
    "status": "[Status of the partial IdP]",
    "publicKey": "[Public Key]",
    "spawnDate": "[Date of Creation]",
    "did": {
        "id": "did:umu:Partial-IdP:X",
        "context": "https://www.w3.org/ns/did/v1",
        "service": {
            "serviceEndpoint": "[Service Endpoint Here]",
            "type": "Partial-IdP"
        }
    }
}
```

Listing 4.1: Partial Identity Provider data model

Once the representation of a partial identity provider has been delineated, we can further explore the concept of the Virtual Identity Provider (vIdP). The vIdP is constructed by amalgamating multiple partial identity providers (pIdPs) under a single management entity, which cryptographically binds each pIdP into the overarching vIdP structure. This integration ensures a cohesive and robust identity management framework. As depicted in Listing 4.2, the structure of the vIdP, akin to that of the pIdPs, varies based on the number of pIdPs integrated. Each pIdP contributes its unique endpoint, ID, and public key. The vIdP itself is distinguished by its own DID ID, for instance, *did:umu.vIdP:1*, along with its status and creation date. Although registered as a tangible entity, the vIdP fundamentally represents the collective composition of all listed pIdPs.

```
{
"spawnDate": "[Date of Creation]",
"status": "[vIdP Status]",
"idps":
    [ "did:umu:Partial-IdP:0",
      "did:umu:Partial-IdP:X"
    ],
"schemas":
    [ "did:umu:PublicParameters:Scheme"],
"did": {
    "@context": "https://www.w3.org/ns/did/v1",
    "id": "did:umu:vIdP:1",
    "services":
    [{
            "endpoint": [Partial IdP endpoint]",
            "id": "did:umu:Partial-IdP:0",
            "pk": [Public Key]"
        },
        {
            "endpoint": [Partial IdP endpoint]",
            "id": "did:umu:Partial-IdP:X",
            "pk": "[Public Key]"
        },
        ...
    ]},
}
```

Listing 4.2: Virtual Identity Provider data model

Lastly, the service registration process uses a similar method as those used for pIdP and vIdP (as illustrated in Listing 4.3), to show the enrolment of a service in the identity framework.

Service enrolment is divided into two primary sections. First is the block DID, where the service creation date and its current state in the system are indicated, like in pIdP and vIdP. Secondly, there is the block of predicates. This last block is paramount since it outlines the data that will be needed for correct operation of the service and how this data will be used. That is to say, every service seeking to operate under the identity system needs to entail a positive statement on the actual private user data that would be used.

All these by tapping into smart contracts and the inherent characteristics of the DLT systems ensure that service behavior is predictable, hence safeguarding against any surprises in how users' private information will be managed.

A typical predicate could be asking the user to confirm his age to let him use the service or to restrict access to the service. Listing 4.4 gives an example of constructing such a predicate. It could, in fact, directly expose the data or perform more complicated checks if the operation defined in the predicate takes place; for example, it could check if the user's age is greater or less than some value, or even between some range values.

This level of flexibility does provide compliance to most of the services, which require verification of certain criteria most of the time. Further, more than one predicate may be registered for a service. This implies services may aggregate different predicates upon system registration in order to achieve more granularity and preciseness of data use and user interaction.

```
{
"date": [Registration date]",
"did": {
    "@context": "https://www.w3.org/ns/did/v1",
    "id": "[Service id]",
    "service": {
        "serviceEndpoint": "[Service endpoint]",
        "type": "[Service type]"
    }
},
"status": "ACTIVE",
"domain": "[Service endpoint]",
"predicates": "[Service predicates data model]"
}
```

Listing 4.3: Service enrolment data model

```
{[
    {
        "attributeName": "url:Age",
        "operation": "GREATERTHAN",
        "lowerBound": 18,
        "upperBound": null
    },
    {
        "attributeName": "url:Age",
        "operation": "REVEAL",
        "lowerBound": null,
        "upperBound": null
    },
    ...
]}
```

Listing 4.4: Service predicate data model

4.5. Conclusions

In the evolving landscape of digital identity management, the necessity for robust, privacy-preserving, and user-centric systems has never been more critical. Traditional

centralized Identity Management (IdM) systems, with their inherent privacy, trust, and security vulnerabilities, are increasingly seen as inadequate for modern digital interactions. Chapter 4 has presented a solution that leverages the Chapter 3 to forge a new path towards decentralization using Distributed Ledger Technologies (DLT) and Enhanced Attribute-Based Credentials (ABC), aiming to rectify these shortcomings.

4.5.1. Core Objectives Revisited

There are three primary objectives in the proposed framework:

1. **Decentralization and Privacy:** Aim to dismantle the monolithic nature of traditional IdM systems by distributing the identity provider's role. This strategy inherently reduces the risk of data breaches and identity spoofing, thereby safeguarding user data more effectively.

2. **Trust Enhancement:** The integration of DLT into our IdM solution is designed to foster a deeper sense of trust among all stakeholders. This technology not only secures transactions and ensures data integrity but also builds a foundation of trust that is critical in the digital age.

3. **User-Centric Control:** In alignment with GDPR mandates, this approach emphasizes empowering users with clear and straightforward mechanisms to manage their personal information, ensuring their privacy is never compromised.

4.5.2. Technological Advancements

Building upon the initial framework presented in Chapter 3, this solution advances the technological foundation in two significant areas:

1. **Distributed Ledger Technologies:** By harnessing the power of blockchain, we enhance the IdM ecosystem's security and trustworthiness. This not only secures transactions but also plays a pivotal role in maintaining the integrity of user identities.

2. **Enhanced Attribute-Based Credentials:** ABCs are meticulously designed to diminish the dominance of any single identity provider. This creates a more equitable and secure system, protecting user identities across various platforms.

4.5.3. Potential Impacts and Challenges

The solution's potential to radically improve digital identity management comes with its set of challenges and impacts, which include:

- **Increased Security and Privacy:** The decentralized nature of our framework significantly reduces centralized points of failure, offering a more resilient approach to digital identity management.

- **Compliance with GDPR:** By enhancing user control over personal data, our solution not only respects user privacy but also aligns closely with GDPR, setting a new standard in data management.
- **Enhanced Trust in Digital Transactions:** The seamless integration of DLT within the IdM ecosystem is poised to elevate trust among users, service providers, and identity providers alike.

However, realizing these benefits is not without its hurdles:

- **Complexity of Implementation:** The deployment of a decentralized framework introduces a level of complexity that could potentially affect scalability and interoperability.
- **Scalability and Interoperability:** Ensuring that our solution scales efficiently while remaining interoperable with existing systems is a daunting task that requires careful consideration and innovative solutions.

4.5.4. Anticipated Drawbacks

The envisioned framework, while promising, is susceptible to several drawbacks:

1. **Performance Overheads:** The incorporation of blockchain and cryptographic methods may introduce latency, particularly in identity verification processes.
2. **Usability Concerns:** Managing cryptographic keys and understanding the nuances of data sharing could potentially hinder the user experience, necessitating user education and streamlined processes.
3. **Adoption Hurdles:** The shift to a decentralized system may encounter resistance due to perceived risks and the significant changes it brings to operational processes.
4. **Regulatory Challenges:** Navigating the decentralized framework within the confines of GDPR and other regulations presents a unique set of challenges, particularly in enforcement and accountability.

Final Thoughts In summary, Chapter 4 presents a forward-thinking approach to redefining identity management in the digital domain. By leveraging the capabilities of DLT and ABCs, we address the critical flaws inherent in traditional systems. However, the journey to realizing the full potential of this framework requires navigating through a maze of technical, usability, and regulatory challenges. Success in this endeavor promises a future where digital identity management is not only secure and privacy-centric but also empowers users like never before.

CHAPTER 5

Implementation and results

5.1. Introduction

To effectively demonstrate the usability of the proposed methods, testing them in real-world scenarios is essential. This chapter focuses on evaluating the proposed solutions through a series of practical use cases, aimed at showcasing how these methods can enhance user privacy and control over their data in contexts requiring digital verification of identity or attributes.

Prior initiatives, such as the ReliAble euRopean Identity EcoSystem (ARIES) [21] and Attribute-based Credentials for Trust (ABC4Trust) [2], have significantly advanced privacy and user protection. However, their widespread adoption encountered numerous obstacles. Subsequent efforts, including Oblivious Identity Management for Private User-Friendly Services (OLYMPUS) [15,17] and Cybersecurity for Europe (CS4EU) [90], have furthered this progress, enhancing the development of sophisticated, privacy-enhanced identity management systems. Notably, the OLYMPUS project achieved the first practical deployment of the distributed identity system detailed in Chapter 3. Almost simultaneously, the CS4EU project faced its own challenges and use cases providing the opportunity to seek an alternative solution by applying the concepts developed in chapter 4. This strategy effectively unifies both projects by integrating their foundational technologies and expanding their potential through the application of the distributed identity framework and distributed ledger technologies.

5.1.1. Overview of the System Architecture

Building on the insights gained from previous initiatives and the challenges identified in Chapters 3 and 4, the proposed identity management system leverages the principles

of privacy-preserving and distributed technologies. The architecture integrates core components of decentralized identity management with advanced privacy features and trust mechanisms enabled by distributed ledger technologies (DLT).

Core Components The system architecture comprises the following core components:

- **Distributed Identity Providers (DIP):** These entities are responsible for managing and verifying user identities in a decentralized manner. By distributing the responsibilities across multiple providers, the system enhances resilience and reduces the risk of a single point of failure.
- **Privacy-Preserving Mechanisms:** Techniques such as zero-knowledge proofs (ZKP) and homomorphic encryption ensure that user data remains private and secure throughout the identity verification process.
- **DLT-Based Trust Framework:** A distributed ledger records identity transactions, providing an immutable and transparent record that enhances trust among users, service providers, and identity providers.
- **User-Centric Control:** Self-sovereign identity (SSI) principles are implemented to give users greater control over their personal data, aligning with GDPR requirements and enhancing user trust in the system.

Implementation Strategy The implementation strategy comprises two main phases, each targeting specific aspects of the system's development and corresponding to different use cases:

- **Phase 1: Non-DLT Enabled, Distributed Identity Provider:** This phase establishes a distributed identity provider without DLT integration, ensuring robust and user-friendly functionalities for user authentication and identity verification through distributed privacy-preserving techniques. It is demonstrated in the first use case, "Pandemic Booking" 5.3.1.
- **Phase 2: DLT Enabled Distributed Identity Provider:** This phase integrates DLT into the distributed identity provider, enhancing trust and transparency by utilizing blockchain technology to create an immutable record of identity transactions. It covers the configuration of the DLT network, interaction between DLT nodes and DIPs, and processes for recording and verifying identity transactions on the ledger. This phase is showcased in the second use case, "Smart City" 5.3.2.

The design and implementation are directly influenced by the concepts and methods from Chapters 3 and 4:

- **Chapter 3 - Privacy-Preserving Distributed Identity Management:** The principles and privacy-preserving techniques discussed in chapter 3 guide the implementation of distributed identity providers and privacy mechanisms, addressing privacy concerns with zero-knowledge proofs and homomorphic encryption.

- **Chapter 4 - DLT-Enabled Identity Management System:** The DLT-based architecture and trust-enhancing features proposed in chapter 4 are integral to the system. The detailed processes and architectural blueprints from chapter 4 provide a framework for incorporating DLT, ensuring a transparent and immutable record of identity transactions that boosts trust and security.

By synthesizing these theoretical insights and architectural designs, we create a robust and scalable identity management system that tackles the key challenges of privacy, trust, and security. The following sections will delve into the practical aspects of this implementation.

5.2. General implementation details

This section delves into the implementation specifics of the two proposals discussed in Chapters 3 and 4. Detailed explanations and technical descriptions are provided to elucidate the practical aspects of applying these proposals.

5.2.1. Non-DLT enabled, distributed identity provider

The framework comprises three primary components: a client Java library, a server component, and an optional verifier component. The server can function as a standalone Java library for custom server applications or be deployed as a self-contained REST-based server.

The client library provides essential methods such as *createUser* and *authenticate*, which facilitate client-server interactions. These methods translate client requests into protocol-specific messages sent to the vIdP servers. The server processes these requests and returns outputs understandable by client applications, thus abstracting the underlying cryptographic protocols used to authenticate users. Although different protocols might generate different types of tokens, which necessitates some awareness by the client of the output format.

The framework supports three main protocols:

- A traditional password verification scheme where the client's password is sent to the server, compared against a stored (salted and hashed) version. If the verification succeeds, a JWT token is generated and returned.

- A distributed scheme based on PESTO [91], engaging at least two distinct servers. It involves Distributed Password Verification (DPV 3.4.3) and Distributed Token

Generation (DTG 3.4.3). Successful DPV leads to a distributed RSA signature on user attributes, combined into a JWT token by the client.

- A scheme using distributed pABC protocols, similar to PESTO for DPV but diverges in the final output, producing a short-lived distributed pABC token instead of an RSA signature. Clients merge these tokens to create a regular pABC that issues attribute-based credentials.

The server component is versatile, deployable as a web server, a servlet, or a library integrated into custom server applications. It handles:

- Protocol endpoints corresponding to the client's needs (traditional password, PESTO, and pABC credentials).
- Management of user attributes, which may be accomplished through authentication protocols alone or augmented by database interactions or other external resources.
- Validation of user claims or identity proofing. This task ensures the authenticity of user attributes before linking them to an account. Implemented modularly, the server supports various identity proofs like X.509 certificates, JWT tokens, or custom eID tokens, facilitated by the *IdProofer* interface. This interface decouples the identity proofing method from the application logic, enabling seamless integration of diverse validation mechanisms.

These server responsibilities are modularized to enhance integration with specific application requirements. The management of user attributes typically requires integration with a database for persistence, potentially involving external data sources. The validation of user claims adapts to the scenario, supporting a broad spectrum of validation methods to ensure flexible and secure user authentication.

The framework integrates with other widely used authentication technologies, mainly: OAUth, OpenID/JWT and W3C Credentials.

The objective of OAuth is to allow a resource owner to grant some service provider access to a protected resource without exposing the user's credentials. The concrete flow can be found in figure 2.7. In particular, the grant type flows are of special interest for our integration, summarized as follows:

1. **User Access Request**: A user seeks access to a service. This service, referred to as the Client, requires the user to verify their identity. To facilitate this, the Client redirects the user to an Identity Provider (IdP) that is authorized to access a restricted resource. The redirection URL includes, among other parameters, an identifier for the service provider and a callback URL, which specifies where the user should be redirected post-authorization and the operations to be authorized.
2. **Authentication and Authorization**: The user follows the redirection to the IdP, where they authenticate and authorize the requested operation. The IdP then issues a token, the type of which depends on the grant type used in the process.

3. **Token Exchange**: In the case of the authorization code flow, the service provider receives a preliminary token from the IdP and uses it to authenticate itself with the IdP.

4. **Access Token Generation**: Upon successful authentication by the service provider, the IdP generates a valid access token.

5. **Resource Access**: The service provider receives the access token, enabling it to access the restricted resource on behalf of the user.

To align with the described authentication flow, a key challenge is managing the redirection to the Identity Provider (IdP). Given that the virtual IdP (vIdP) consists of multiple servers, the redirection process must encompass all these servers. Moreover, the client is required to engage in a cryptographic protocol, making it insufficient for a standard browser to merely follow the URL. On the upside, the flexibility of the vIdP allows for the generation of a diverse range of tokens. Once the redirection issue is resolved, the entire flow can be effectively implemented using a vIdP.

It is important to note that the authorization code grant flow involves issuing a temporary token. The Service Provider must subsequently contact the vIdP to exchange this temporary token for a proper access token. While theoretically possible, this exchange necessitates that the Service Provider amalgamate distributed signatures into a single access token. This additional requirement may complicate the Service Provider's application and could impede broader adoption. Additionally, the authorization code flow presents challenges in maintaining user-service provider traceability anonymity from the IdP. Due to the complexities mentioned, the framework does not support the authorization code flow. Instead, it utilizes the implicit flow, which streamlines the authentication process. Once the user is authenticated in the browser (or a cookie is set), the system automatically handles credential transmission. However, this approach has a limitation: the browser indiscriminately sends the credential to any application that requests it from the "authenticated domain." As a result, only clients within the same domain as the server can successfully communicate, leading to potential loss of control over where credentials are sent.

The challenge of coordinating multiple servers in a virtual IdP setup is addressed by deploying a client application that runs locally on the user's machine. This application provides a local OAuth IdP REST interface, offering functionality akin to that of a traditional IdP. The service provider generates a redirect URL targeting the client's localhost. When the client's browser follows this redirect, it connects to the locally hosted application, which then communicates with the virtual IdP servers. This arrangement makes the underlying virtual IdP-based protocols transparent to the user. Moreover, this solution is versatile enough to be adapted for use with other authentication schemes, such as SAML. However, a significant practical challenge is the requirement to install custom software on the client's device.

In theory, the issue of installing custom software on client devices could be mitigated by running the software on a trusted server or by deploying a JavaScript application

that the service provider links to. However, both alternatives introduce a single point of trust. This arises because either the user's credentials must be transferred away from the client device, or the user must trust the JavaScript provided by the server.

The OpenID integration is built on top of OAuth 2.0 [28] and is used for user authentication rather than authorization. Rather than having the IdP authorize the user's access to some resource, OIDC uses the IdP to authenticate the user and lets the Service Provider handle the authorization. While both OAuth implicit grant and authorization code flows can be used in connection with OIDC, only implicit grant is supported by the framework.

In addition to leveraging OAuth, OpenID Connect (OIDC) utilizes JSON Web Tokens (JWT) for representing user identity [92]. JWTs are compact and self-contained mechanisms for securely transmitting information as JSON objects. These tokens encapsulate a set of claims and support both signatures and encryption. Each token consists of three parts: a header that specifies the encryption or signature algorithms used, a payload containing the JSON data, and a signature. For efficient transmission via HTTP(s), the entire structure is Base64 URL-encoded. The following JSON, listing 5.1, is a sample of a JWT token issued by vIdP *vidp.umu.eu*, to the service provider *restaurant-provider* attesting the username attribute *alice*.

```
{
    "sub":"alice",
    "iss":"https://vidp.umu.eu",
    "aud":"restaurant-provider",
    "auth_time":1311280969,
    "iat":1311280970,
    "exp":1311281970
}
```

Listing 5.1: vIdP issued JWT token example

In addition to the *sub* (subject or username) attribute, OIDC defines a set of standard user attributes or claims, including *email*, *birthdate*, *name*, and others. As OIDC is built upon OAuth, it inherits both the challenges and the solutions associated with the OAuth framework. The PESTO protocol, which provides a generic signature scheme, facilitates the generation of distributed RSA signatures on JWT tokens. These signatures can then be recombined efficiently. Notably, JWT is the default output format of the PESTO protocol implementation.

The integration of W3C Verifiable Credentials [69] with the PESTO protocol is straightforward, primarily involving the construction of the appropriate JSON structure for signing. The potential inclusion of domain and challenge attributes within a linked proof could offer significant enhancements to the PESTO protocol, particularly in terms of privacy. Such improvements aim to reduce linkability by incorporating mechanisms to prevent the misuse of valid JWTs across different relying parties in OIDC scenarios. Specifically, the JWT includes an *aud* (audience) attribute, which

delineates the intended recipient of the token. This requirement compels the Identity Provider (IdP) to recognize the relying party, thereby complicating efforts to create an unlinkable IdP solution that complies with OIDC standards, as the IdP must be aware of the relying party's identity.

Although extensive research is still pending, our preliminary contributions are detailed in the publication *Towards a Standardized Model for Privacy-Preserving Verifiable Credentials* [18]. This work explores the feasibility of IdPs issuing verifiable credentials that maintain unlinkability, further advancing the discussion on privacy-preserving digital identity management.

A major barrier to the adoption of pABC systems has been the challenge of integrating them with existing systems and other pABC schemes. The W3C Verifiable Credential specification, which is still under development and subject to changes, presents integration challenges. Certain functionalities remain unimplemented due to time constraints.

In an effort to facilitate the integration of our system with existing infrastructures, we have adapted our data model to align with the W3C Verifiable Credential specification [93]. This adaptation required us to define the usage profile of the standard, specifically identifying which optional functionalities to implement and establishing definitions for constructs unique to our methodology.

We consider the inclusion of a **context** field in every credential and presentation, which must encapsulate the VC context, the overarching context of the project, and a deployment-specific context addressing the relevant attributes. The project context comprehensively defines the OLYMPUS credential and presentation types, along with three novel types of cryptographic proofs essential for our operations. The first proof type, *OlPsSignature*, listing 5.2, legitimizes a credential, while the subsequent types, *OlPsDerivedProof*, listing 5.3 and *OlPsDerivedProofRange*, listin 5.4, are used for deriving proofs in presentations.

```
"proof": {
    "type": "OlPsSignature",
    "epoch": 17801571234,
    "proofValue": "eyJraWQiOI29seW1w..dXMjMTIzNCkCIsImFs",
    "proofPurpose": "assertionMethod",
    "verificationMethod": "did:example:vIdP"
}
```

Listing 5.2: W3C OlPsSignature

```
"proof": {
    "type": "OlPsDerivedProof",
    "proofValue": "eyJraWQiOiJkaWQ6bWV0YTpURVNUI29seW1w..dXMjMTIzNCIsInR5
        cCI6IkpXVCIsImFs",
    "verificationMethod":"did:example:vIdP",
    "nonce": "randomMessageSignedEstablishedInPolicyExchange",
    "epoch": 17801571234,
    "proofPurpose": "assertionMethod"
}
```

Listing 5.3: W3C OlPsDerivedProof

```
"proof": {
    "type": "OlPsDerivedProofRange",
    "proofValue": "eyJraWQiOiJkaWQ6bWV0YTpURVNUI29seW1w..dXMjMTIzNCIsInR5
        cCI6IkpXVCIsImFs",
    "verificationMethod":"did:example:vIdP",
    "nonce": "randomMessageSignedEstablishedInPolicyExchange",
    "epoch": 17801571234,
    "proofPurpose": "assertionMethod",
    "rangeProofs": [
        { "attr": "height",
          "commitment": "Ekjd1...12==",
          "lowerBoundProofValue": "asd...1a2==",
          "upperBoundProofValue": "asd...1a2=="
        }]
}
```

Listing 5.4: W3C OlPsDerivedProofRange

For all these proofs, we intend to have the verification method be an URL to a *verificationMethod* JSON object with the necessary information for verification, listing 5.5, inspired in the DID core standard [63].

```
{
    "type": "OlPsSignatureVerificationKey",
    "id": "did:olympus:vIdP#aggKey",
    "publicKeyBase64": "QDui4QA6QKMpMRpJQxm8TUV...iZRu6aNPgmERrXUPBo8hc",
    "curve": "BLS461",
    "attributes": "https://deployment.com/example/definitions"
}
```

Listing 5.5: W3C Verification Method

As each use case will define attributes tailored to its needs, use cases will need to create a context (to be included along the framework general context), listing 5.6

In addition, it is crucial to establish a common and standardized framework for the operations we can perform. Central to this framework is the representation of

```
"@context": [
    {"@version": 1.1},
    "https://www.w3.org/ns/odrl.jsonld",
    {
        "ex": "mDL-olympus-deployment.com/example/",
        "schema": "https://schema.org/",
        "givenName": "ex:givenName",
        "height": "ex:nationality",
        "dateOfBirth": "ex:birthDate",
    }
]
```

Listing 5.6: W3C Use case context

predicates as simple JSON objects. In presentations, attributes (JSON properties) can be expressed in two forms: either as a valid value or as a predicate. Each predicate is characterized by an *operation* tag, which delineates the relationship between the attribute being proved and its *value*. The internal structure of the *value* varies depending on the specific operation, as illustrated in Listing 5.7.

```
{
    "operation": "inRange"
    "value": {
        "lowerBound": 0
        "upperBound": 10
    }
}
{
    "operation": "le"
    "value": {
        "lowerBound": 10
    }
}
```

Listing 5.7: W3C Predicate definition

Tags such as *"ge"*, *"le"*, and *"inRange"* represent the relational predicates *greater-or-equal*, *lesser-or-equal*, and *between-values*, respectively. For these tags, the *value* property should include appropriate boundaries: *"lowerBound"* and *"upperBound"* for *"inRange"*; the respective boundary values for *"ge"* and *"le"*. Conversely, tags like *"memberOf"* and *"nonMemberOf"* indicate proofs of *membership* and *non-membership*. Here, the *value* must encompass a *"set"*. Within our data model, the *"set"* property can either be a direct enumeration of set values or more usefully, a URL that defines the set externally. This latter approach is particularly advantageous in systems requiring public parameters or specific setups for each set, facilitating external reference and validation.

Upon defining all necessary structures in accordance with the W3C standard, we can generate a W3C verifiable credential as shown in Listing 5.8. This credential adheres strictly to the standard, specifying the requisite contexts for proper validation. It includes the credential type, the schemas needed for validation, and their locations. Additionally, it encompasses typical credential data such as the issuer, date of issue, expiration date, attributes, and the cryptographic proof. In the same way, we can generate W3C verifiable presentations based on the credential obtained, as shown in the listing 5.9.

```
{
    "@context": ["https://w3id.org/credentials/v1",
        "https://olympus-project.eu/context",
        "https://example-olympus-deployment.com/context"
    ],
    "type": ["VerifiableCredential","OlympusCredential"],
    "credentialSchema": [
        {
            "id": "https://example-olympus-deployment.com/schemas/
                validationSchema",
            "type": "OlZkValidationSchema"
        },
        {
            "id": "https://olympus-project.eu/example/encodingSchema",
            "type": "OlZkEncodingSchema"
        }
    ],
    "issuer": "did:example:OL-vIdP",
    "issuanceDate": "2021-06-07T18:13:16",
    "expirationDate": "2021-06-08T14:13:16",
    "credentialSubject": {
        "familyName": "Doe",
        "givenName": "John",
        "dateOfBirth": "1980-03-06T00:00:00",
        "height": 185
    },
    "proof": {
        "type": "OlPsSignature",
        "proofValue": "CjwKOgAAAAAAAAAAA...6E=",
        "epoch": 1623161596000,
        "verificationMethod": "did:example:OL-vIdP:method",
        "proofPurpose": "AssertionMethod"
    }
}
```

Listing 5.8: W3C Credential

```
{
    "@context": [...],
    "type": ["VerifiablePresentation","OlympusPresentation"],
    "expirationDate": "2021-06-07T18:14:20",
    "verifiableCredential": [
        {
            "credentialSchema": [...],
            "credentialSubject": {
                "givenName": "John",
                "dateOfBirth": {
                    "operation": "le",
                    "value": { "upperBound": "2003-06-07T00:00:00" }
                }
            },
            "issuanceDate": "2021-06-07T18:13:16",
            "issuer": "did:example:OL-vIdP",
            "type": ["VerifiableCredential","OlympusCredential"],
            "expirationDate": "2021-06-08T14:13:16",
            "proof": {
                "type": "OlPsDerivedProofRange",
                "proofValue": "CvMBCvA...w==",
                "epoch": 1623161596000,
                "rangeProofs": [
                    {
                        "attr": "https://example-olympus-deployment.com/
                            attributes/DateOfBirth",
                        "commitment": "CngK...BMZs=",
                        "lowerBoundProofValue": "CnoKe...zPQDlNtX",
                        "upperBoundProofValue": "CnoKe...AJm"
                    }
                ],
                "nonce": "signedMessage",
                "verificationMethod": "did:example:OL-vIdP:method",
                "proofPurpose": "AssertionMethod"
            }
        }
    ]
}
```


Listing 5.9: W3C Presentration

Continuing with implementation details, for a successful vIdP deployment, it is essential to integrate, configure, and expose the various components of the framework to the client. Typically, the integration of components is conducted in Java, with each partial Identity Provider (IdP) functioning as a Java program. The methods from these programs are made accessible to the client through a REST interface, which aligns with the expectations of the default client software. This REST interface can be implemented and deployed in several ways to suit the specific needs of the application.

In addition to integration, configuring each partial IdP is crucial; this includes

setting the ports for exposing the REST interface, managing communication with other partial IdPs, and configuring key management protocols. The framework includes configuration interfaces such as *ServerConfiguration* and *PABCConfiguration*, which outline important parameters for DTG and DCI protocols, allowing for flexible representation of configurations in each deployment.

For the user client, implementations may vary; they typically require parameters such as a list of the partial IdPs and an implementation of the *ClientCryptoModule* during setup. In scenarios involving pABC, a *CredentialManagement* component is also necessary. This component handles the storage and management of Attribute-Based Credentials (ABCs), deciding whether credentials should be retained while valid or discarded after the derivation of a presentation token.

Authentication processes yield an access token, the specifics of which depend on the user client implementation. For example, in the PESTO scenario, a JWT token may be issued, whereas a pABC implementation might generate a presentation token. The generation of this token also varies; in PESTO, an authentication protocol is executed with the vIdP, leading to the creation of a signed access token if the protocol is successful. In contrast, in the pABC framework, a valid credential stored locally can enable the generation of the access token without further interaction with the vIdP. The policy parameter defines the content of the access token, specifying a list of predicates that dictate which attributes should be revealed or compared. For instance, if a policy requires the attributes ”name” and ”age” to be disclosed, the access token will contain a vIdP-attested proof of these attributes.

Regarding the verifier deployment tokens generated using the PESTO approach (both the original and the ones adapted for the OIDC flow) are standard JWTs and can be validated with any JWT verifier (or OIDC verifier) without using project-specific code. The pABC approach necessitates custom software to manage cryptographic operations, such as the verification of presentation tokens. The framework includes a Java-based library specifically designed for this purpose. The primary interface for verification is *PABCVerifier*, and for use cases involving W3C VC serialization, we provide *W3CPresentationVerifier* with a corresponding method:

- **VerificationResult:** This method accepts a token (serialized as a `String`) and an OLYMPUS policy.

The *VerificationResult* is an enumeration with the following possible outcomes:

- **VALID**: The token has been validated and fulfills the policy.
- **INVALID_SIGNATURE**: The cryptographic operations, including signatures and Zero-Knowledge proofs, have failed.
- **BAD_TIMESTAMP**: The token or credential has expired.
- **POLICY_NOT_FULFILLED**: The token does not meet the requested predicates.

The verification library relies minimally on non-standard libraries, except for a library supporting Bilinear pairings where we also use Apache Milagro AMCL [94]. Although the verification process is typically performed by a relying party on a common "server", adapting the library for use in other environments should be straightforward.

In conclusion, this framework marks a significant advancement in digital identity management, integrating versatile identity verification methods such as pABC and PESTO with established standards including OAuth, OpenID Connect, and W3C Verifiable Credentials. By effectively addressing both practical and theoretical challenges, it offers a scalable, secure, and flexible solution suited for a variety of deployment scenarios. In 5.2.2 , we will delve into enhancements to this framework, particularly focusing on the incorporation of distributed ledger technologies as proposed in Chapter 4.

5.2.2. DLT enabled distributed identity provider

Chapter 4 presents an evolution of the distributed identity provider framework that primarily incorporates interactions with Distributed Ledger Technologies (DLT), smart contracts, and introduces a new entity known as the blockchain handler.

The foundational aspects of the distributed identity framework remain unchanged; therefore, this section will focus on the implementation of DLT, smart contracts, and the integration of the new entity.

The implementation is structured as follows. An Android-based user client, three JAVA-based partial IdPs, a Javascript-based Blockchain Handler entity and a blockchain platform.

From 5.2.1, we understand that developing the user client is straightforward, requiring only the integration of OLYMPUS dependencies and minor code modifications to adapt to the new scenario. Conversely, the partial Identity Providers (IdPs) must integrate functionality to natively communicate with the blockchain platform. The specifics of this integration can differ significantly depending on the chosen Distributed Ledger Technology (DLT) platform. Additionally, the implementation of the handler will also need to be adjusted to align with the processes involved.

Selecting the right blockchain platform for integration is a complex decision. Our review included various options within the Hyperledger ecosystem [62]. We initially experimented with the Hyperledger Indy project [95] due to its specialization in identity management. However, our initial tests revealed certain limitations, notably Indy's insufficient support for smart contracts [96,97], which proved to be a critical shortcoming. While Indy is adept at handling identity-related operations such as creation, storage, and verification of digital identities, it does not support the execution of complex business logic through smart contracts. This is a stark contrast to other platforms that offer robust smart contract capabilities, enabling the automation and processing of a wide range of transaction types beyond mere identity assertions. The lack of such functionalities in Indy is a considerable drawback for applications that demand dynamic transactional logic and extensive automation, typical of more advanced smart contract environments. Consequently, the absence of a comprehensive smart contract

infrastructure in Indy poses significant challenges in implementing the dynamic and responsive interactions that are crucial to achieving our project objectives.

After evaluating different platforms, we considered Hyperledger Fabric [98], which stands out due to its comprehensive suite of features, including a robust implementation of smart contracts and a highly modular architecture. Although not primarily designed for identity management, Fabric provides a degree of flexibility and a broad spectrum of functionalities that are well-suited to our project's needs.

Fabric's architecture is particularly acclaimed for its low latency and strong privacy controls, essential for applications requiring secure and rapid transaction processing. Its modular design allows for significant customization, enabling organizations to tailor the blockchain precisely to their operational requirements. This can include varying consensus mechanisms, specialized membership services, and unique privacy provisions that enhance transactional confidentiality and user anonymity.

The scalability and efficiency of Fabric are further highlighted by extensive empirical studies [99, 100]. These studies confirm Fabric's capability to efficiently handle high throughput, processing up to 200 transactions per second, and supporting a network that exceeds 100,000 participants. Such performance is critical for large-scale deployments and is indicative of the platform's robust infrastructure and optimized processing capabilities.

Moreover, Fabric has demonstrated exceptional performance in operational benchmarks, with an average response time of just 0.01 seconds across 100,000 "query" transactions [101]. This performance metric is particularly relevant in environments where quick data retrieval is crucial, ensuring that even under substantial load, the system remains responsive and efficient.

The comprehensive feature set, coupled with proven scalability and performance metrics, makes Hyperledger Fabric an ideal choice for our end. Its ability to be configured extensively for specific application needs—while maintaining high standards of security and efficiency—positions it as a superior choice for integrating advanced blockchain solutions into diverse and demanding operational landscapes.

In Hyperledger Fabric, smart contracts are referred to as *Chaincodes*, which play a pivotal role in the platform's architecture. Chaincodes encapsulate and implement the business logic of transactions, automating and executing complex contractual behaviors directly on the blockchain. Crucially, Chaincodes are intricately linked with another fundamental component of Fabric—*channels*.

In Hyperledger Fabric, channels serve as private sub-networks that enable secure and confidential communications among specified network members. These channels facilitate the execution of private transactions, effectively isolating them from the broader network. Such isolation ensures that sensitive data and transaction details remain exclusively accessible to authorized participants, significantly enhancing transaction security and privacy by tailoring access based on the specific needs and permissions of the network participants.

Channels incorporate a set of chaincodes that govern the operations and business logic particular to each channel. Figure 5.1 illustrates this configuration, showing three

different organizations participating in three distinct channels, each operating with its own set of chaincodes.

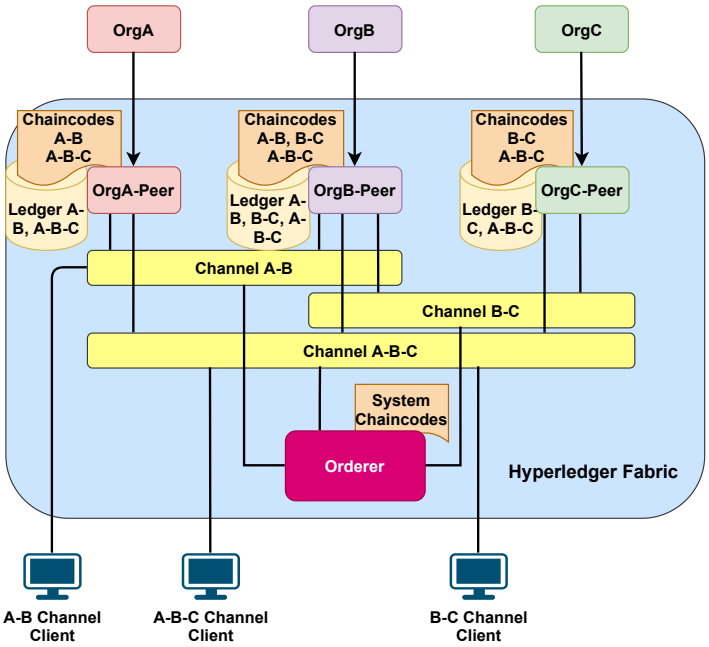


Figure 5.1: Hyperledger fabric channels

The blockchain deployment within Fabric is structured around various key components:

- **Organizations or Members**: These are the entities or consortia members participating in the blockchain network, each with specific roles and permissions.
- **Anchor Peers**: Designated by each organization, anchor peers act as a communication bridge for disseminating transaction data and blockchain state information within their organization.
- **Ordering Node (Orderer)**: A critical component that maintains consistency and sequences transactions across the blockchain, packaging them into blocks and ensuring their delivery to all network peers.
- **Chaincodes**: These smart contracts define the transactional logic and are executed on the blockchain to manage the ledger state.
- **Shared Ledger**: A fully replicated ledger that records all transactions across the network, maintained across all peers to ensure transparency and immutability.

In Hyperledger Fabric, each transaction is executed within a designated channel, ensuring that interactions are confined to a controlled and secure environment. All parties involved in a transaction must undergo a rigorous authentication and authorization

process before participating. This stringent requirement guarantees that only verified and authorized entities can perform transactions on a given channel, thereby enhancing security and maintaining the integrity and confidentiality of the data exchanged.

Each channel operates as an independent ledger, maintaining a comprehensive history of transactions visible only to its members. Authentication is typically managed through digital certificates, while authorization leverages Membership Service Providers (MSPs). MSPs define the roles and privileges of network participants, ensuring transactions are transparent among authorized users and protected from unauthorized access. This dual mechanism adheres to stringent security standards and meets compliance requirements, thereby safeguarding the network.

Additionally, in Hyperledger Fabric, chaincodes are instrumental in extending our virtual Identity Provider (vIdP) and partial Identity Providers (pIdP) framework. vIdPs are conceptualized as an aggregation of partial IdPs, each endorsed on the ledger through smart contracts. Each partial IdP begins its operation by invoking an enrollment contract, which captures essential information required for subsequent identification. This includes a DID document detailing an identifier (e.g., `did:umu:OL-Partial-IdP:0:test1`), context, and the service definition, which encompasses the service address and type. Other recorded attributes include the operational status, the spawn date, and the associated public key. Figure 4.5a illustrates this process in detail, and Listing 5.10 provides an example of such an enrollment:

```
{
    "status": "ACTIVE",
    "publicKey": "CnoKeAo6DeVv7T9T[...]",
    "spawnDate": "2021-03-10T10:48:20",
    "did": {
        "id": "did:umu:OL-Partial-IdP:0",
        "context": "https://www.w3.org/ns/did/v1",
        "service": {
            "serviceEndpoint": "10.1.6.6:9080",
            "type": "OL-Partial-IdP"
        }
    }
}
```

Listing 5.10: Partial IdP enrollment

Simultaneously, as partial IdPs are enrolled, the composition of the vIdP to which they belong is also dynamically updated, ensuring no single partial IdP can act as a controller or hold undue influence. This mechanism prevents any hierarchical structure within the vIdP, maintaining equal responsibility among all partial IdPs. The virtual IdP is thus a composite entity made up of the endpoints, DIDs, and public keys (including an aggregated public key) of its constituent partial IdPs, as shown in Listing 5.11:

```
{
    "did":
    {
        "@context": "https://www.w3.org/ns/did/v1",
        "id": "did:umu:OL-vIdP:test1",
        "services":
        [
            {
                "endpoint": "10.1.6.6:9080",
                "id": "did:umu:OL-Partial-IdP:0",
                "pk": "CnoKeAo6er2OxSH2lrVv7T9T[...]"
            },
            {
                "endpoint": "10.1.6.6:9081",
                "id": "did:umu:OL-Partial-IdP:1",
                "pk": "U8R21sGxIE9UebXNMISCdWaZ[...]"
            },
            {
                "endpoint": "10.1.6.6:9082",
                "id": "did:umu:OL-Partial-IdP:2",
                "pk": "aYVXzQ2qNYiJdAgBbPHzYAKA[...]"
            }
        ]
    },
    "docType": "VIdPRegistration",
    "idps":
        [
            "did:umu:OL-Partial-IdP:0",
            "did:umu:OL-Partial-IdP:1",
            "did:umu:OL-Partial-IdP:2"
        ],
    "schemas":
        [
            "did:umu:OL-PublicParameters:Scheme"
        ],
    "spawnDate": "2021-03-10T10:14:44",
    "status": "ACTIVE"
}
```


Listing 5.11: Virtual IdP enrollment

This structured approach ensures that any IdP and vIdP engaged in the network architecture is introduced by a trusted entity, which possesses the necessary permissions and cryptographic materials to operate within the ledger. This setup leaves a traceable and auditable record, enhancing the security and verifiability of the network.

Service Providers play a pivotal verification role within our system. They communicate access policies and validate access requests against these policies. Our innovative approach further augments this functionality by incorporating a registration process for service providers through smart contracts, as illustrated in Figure 4.5b. When a service provider decides to adopt this new methodology, it must register essential details on the ledger: its endpoint, DID, registration date, status, and a set of predicates that define

the necessary data for its operations. For example, these conditions might include revealing an email address or verifying that a user's age falls within a specified range, detailed in Listing 5.12. This proactive registration enables all network participants to identify which services are part of the framework and understand the specific data requirements of each service in advance.

Furthermore, this registration process empowers service providers with the capability to autonomously configure their specific settings related to the virtual Identity Provider (vIdP), enhancing operational flexibility. This dual benefit not only streamlines setup procedures but also significantly enhances adaptability for service providers, as depicted in Figure 4.6.

The ledger serves as an authoritative registry, capturing this crucial information in a manner that is immutable and auditable. Currently, the service registration process is conducted manually, necessitating administrative intervention. However, there is potential for this process to be automated in the future, which would streamline operations and reduce administrative overhead.

```
{
    "date": "2021-03-10T11:28:52",
    "did": {
        "@context": "https://www.w3.org/ns/did/v1",
        "id": "service",
        "service": {
            "serviceEndpoint": "https://myservice.com",
            "type": "Web service"
        }
    },
    "domain": "https://myservice.com",
    "predicates": "[{\"attributeName\":\"url:Role\",\"operation
       \":\"REVEAL\",\"value\":null,\"extraValue\":null}]",
    "status": "ACTIVE"
}
```

Listing 5.12: Service enrolment

In Chapter 3, the user, our primary concern, is safeguarded through the principles of minimal disclosure and the distribution of cryptographic material. This system equips the client with the ability to connect to the ledger, facilitating the discovery of registered virtual Identity Providers (vIdPs) and Identity Providers (IdPs) securely before any registration on the platform. Similarly, legitimate service providers and the specific data requirements they have are also identifiable, placing the user in a strategically advantageous position. From the outset, users can make informed decisions without compromising their privacy or security.

The benefits to the user from this system are threefold:

1. **Trustworthy Connection Configuration:** The initial connection setup is sourced from the ledger, a reliable authority that ensures the configuration process's integrity. This efficiency not only simplifies the setup for end-users but also potentially eliminates the need for manual configuration altogether.

2. **Informed Decision-Making:** Users receive verified information directly from the ledger about available services, including the type of service and the specific data requirements. This enables users to make informed decisions based on objective, ledger-certified data, ensuring they can confidently interact with services, secure in the knowledge that their Identity Provider is reliable and that service offerings are under continual surveillance.

3. **Auto Client Configuration:** Similar to service providers, user applications can perform an auto-configuration process, as illustrated in Figure 4.7. This feature further enhances user experience by simplifying the initial setup and ongoing maintenance of client settings.

Moreover, if a service unilaterally changes its access policies to become more invasive, the ledger's monitoring capabilities ensure that the user is promptly informed of such changes. This system not only empowers users with control over their digital interactions but also maintains a high level of transparency and security.

It is important to emphasize that the ledger does not record any sensitive information, thereby safeguarding against potential tracking or data leakage. This design principle ensures that privacy is maintained throughout the system's operation. While the user client has the capability to interact with the ledger, such as sending alerts about suspicious services, its primary role is observational. This means that while it can contribute to the security of the system by reporting anomalies, it does not actively modify or store sensitive user data on the ledger. This approach not only enhances user privacy but also strengthens the overall security framework by enabling a proactive response to potential threats without compromising the confidentiality of user information.

To conclude, the newly introduced Blockchain Handler entity serves as a REST API that facilitates interactions with the deployed ledger using standard HTTP methods. Developed in JavaScript, this API offers a lightweight and easily extendable framework for ledger interactions. Beyond basic ledger operations, the Blockchain Handler is also designed to simulate various elements of the blockchain ecosystem. This includes functionalities like mimicking a service provider and its corresponding verifier, thereby providing a versatile and practical tool for development and testing.

The advantages of employing the Blockchain Handler are multifaceted:

- **Accessibility:** Being based on REST, it integrates seamlessly with existing web infrastructure, making it accessible to developers familiar with web technologies without requiring specialized blockchain knowledge.

- **Flexibility:** The JavaScript-based implementation ensures that the handler is not only lightweight but also adaptable to various use cases, facilitating quick modifications and enhancements.

- **Testing and Simulation:** The ability to simulate service providers and verifiers within the handler allows developers to conduct thorough testing and scenario

analysis before full-scale deployment. This feature significantly reduces the potential for errors in live environments and improves overall system reliability.

- **Rapid Development:** The easy-to-use interface accelerates development cycles, enabling faster rollout of new features and improvements.

In essence, the Blockchain Handler optimizes the development process by providing robust, user-friendly tools for interacting with and testing the blockchain environment, ultimately enhancing the efficiency and security of deployments.

5.3. Use cases

This section presents two use cases that demonstrate the practical application of the proposals outlined in the preceding sections.

First, the use case titled "The Pandemic Booking" is presented. This use case explores the implementation of the solution introduced in Chapter 3, as well as its potential areas for improvement.

Next, the "Smart City" use case is introduced, in which the solution proposed in the previous use case is extended with the enhancements introduced in Chapter 4. This section highlights the new architectural features, entities, and results obtained.

5.3.1. The pandemic booking

Introduction

This use case explores the unexpected global upheaval caused by the COVID-19 pandemic in 2020, a significant health crisis documented by the World Health Organization[1]. This unprecedented challenge necessitated a thorough reevaluation of various aspects of everyday life, emphasizing particularly the importance of identity management and personal attributes during public health emergencies. As societies around the world struggled with the immediacy of the crisis, the imperative for robust privacy protection mechanisms intensified, enabling individuals to protect their personal data against misuse amid increased surveillance measures.

Impact on Service Operations

The pandemic profoundly transformed the operational frameworks of essential services including healthcare, travel, and education, with a specific focus on the management of personal interactions through booking systems. Organizations across these sectors were required to swiftly adapt to new health and data privacy regulations, catalyzing a substantial rise in demand for secure, resilient digital infrastructures capable of ethically integrating and handling sensitive health data while safeguarding individual privacy.

[1] `https://www.who.int/en/health-topics/coronavirus`

Specific challenges in Data Management

Throughout the pandemic, entities encountered significant challenges:

- Maintaining the accuracy and confidentiality of health data, which became critically important as mismanagement could lead to severe consequences for individual privacy and public health.
- Adhering to rapidly changing legal and health regulations which required agile adjustments to existing systems and operations.

The integration of real-time health data into booking systems necessitated sophisticated technological enhancements to ensure system robustness and reliability.

Innovative Solutions

In response to these challenges, several innovative technological solutions were adopted:

- **Contactless Interfaces:** Developed to minimize physical contact and reduce the risk of virus transmission, these interfaces facilitated safer interactions in public spaces and service areas.
- **Real-Time Health Tools:** From monitoring individual vaccination statuses and thermal temperature readings via infrared cameras to incorporating continuous disinfection systems in public spaces, a variety of systems were deployed during this period. These implementations were crucial in enhancing public safety by ensuring environments were safe for occupancy and reducing the spread of the virus. Each system played a vital role in the comprehensive public health response to the pandemic, demonstrating the importance of integrated health safety technologies and moreover, the importance of data awareness and privacy options.
- **Advanced Encryption Methods:** Implemented to bolster data security, these methods ensured that personal and health-related information was encrypted and securely transmitted across platforms.
- **Comprehensive Software Solutions:** Created to streamline information management, these solutions ensured that all operations remained compliant with evolving health regulations while prioritizing user privacy.

These enhancements not only addressed the immediate operational challenges posed by the pandemic but also laid the groundwork for future technological innovations in managing health crises. The demonstrated effectiveness and adaptability of these solutions highlight the potential for transformative advances in public health infrastructure. Within this evolving landscape, the relevance of the solution proposed in Chapter 3 becomes particularly salient. This proposal is designed to develop systems that are not only more resilient and efficient but also exceptionally secure in handling sensitive user

data, ensuring compliance with standards or regulations (i.e., [3, 69]) and respect for user privacy.

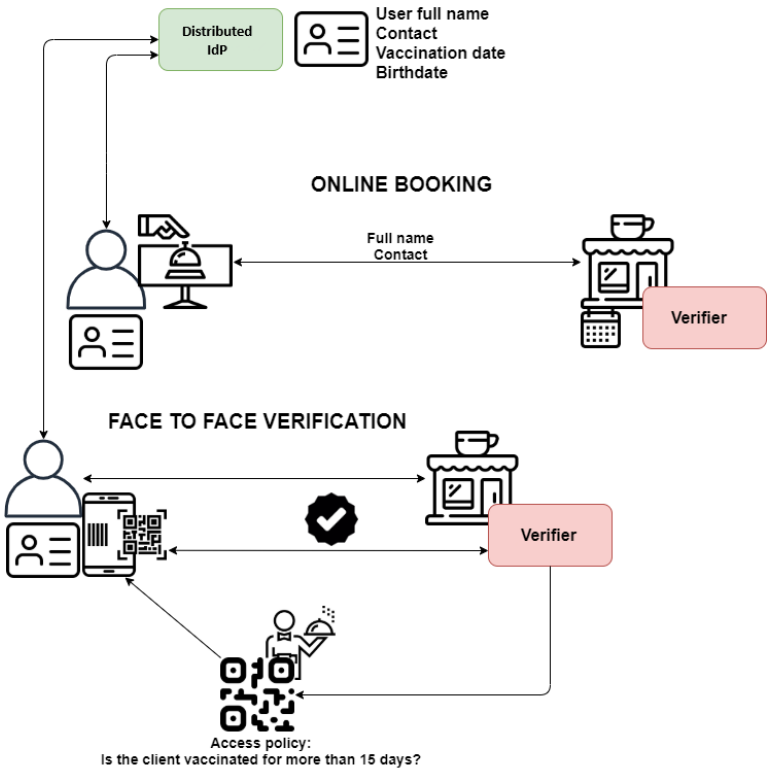


Figure 5.2: Booking scenario

The scenario depicted provides a comprehensive framework for the full implementation of the architecture shown in Figure 3.4. Additionally, it integrates the two workflows described in Chapter 3, which are essential for ensuring a seamless interaction between all parties involved. This structured approach not only enhances security and efficiency but also supports robust identity management across various platforms.

The components involved in the scenario are:

- **Distributed IdP (vIdP):** This component offers privacy-preserving identity management to users while ensuring the security of the information presented to relying parties. It supports online authentication following the OIDC [26, 34] flow and issues pABCs (serialized according to W3C's Verifiable Credentials specification) for "offline" interactions. Optionally, an external service can act as a third-party identity provider for user enrollment in the vIdP. Technically, the vIdP operates as a 3-server deployment of the combined IdP class, integrating both pABC and OIDC functionalities.

- **User:** Two tools are essential, one for each approach: A mobile application for "offline" pABC presentations, and a locally hosted REST-based client, accessible

via a web browser, which may eventually be replaced by a browser plugin. The local REST client not only facilitates OIDC functions but also offers account management features, such as account creation and password changes.

- **Relying-party:** In this scenario is a single "service provider organization" (the restaurant) which operates differently based on the flow. This might involve two distinct functions:

  1. A booking service that facilitates online interactions through a standard OIDC implementation.
  2. An offline waiter function that uses a mobile device with the distributed identity provider verification library to validate pABC presentations upon the guest's arrival.

  The service provider is implemented as a simple NodeJS application, using standard components and libraries wherever possible. OIDC support is provided by redux-oidc, a popular NodeJS [102] library, which allows seamless integration with both vIdP and KeyCloak [103] IdP. For handling pABC presentations, the application also runs a small REST server that exposes two methods: one for verification and another for setup, which configures the endpoints of the partial IdPs that comprise the vIdP for token verification.

While the scenario involves a single overarching flow (booking and making use of a service), it can be divided into two separate sub-processes: (1) **The online reservation** (figure 5.3) and the (2) **in-person check-in** (figure 5.6).

The online reservation system allows users to make a reservation in their name by utilizing the OpenID Connect (OIDC) authentication flow, as depicted in Figure 2.8. The process begins when a user visits the restaurant's webpage (Service Provider, SP) using a standard web browser. On the webpage, the user opts to log in using OLYMPUS, a distributed Identity Provider (DIP). This action directs the user to a login interface that requires a username and password.

Upon entering their credentials, the system initiates the detailed authentication process represented in Figure 3.5, distributed password verification (DPV). Once the authentication is successful, the system generates an OIDC token via the distributed token generation (DTG) protocol, as outlined in Figure 3.6. This token, which includes attributes verifying the user's age (e.g., being over 18 years old), is transmitted through the browser to the SP.

Finally, the SP verifies the authenticity and validity of the OIDC token and, upon successful validation, confirms the reservation. The overall authentication and reservation process is succinctly illustrated in Figure 5.3 and 5.4.

One significant advantage of this OIDC-based approach is its familiarity to users, who often employ this type of authentication in various online interactions, albeit unconsciously. Additionally, from the perspective of the service provider, integrating

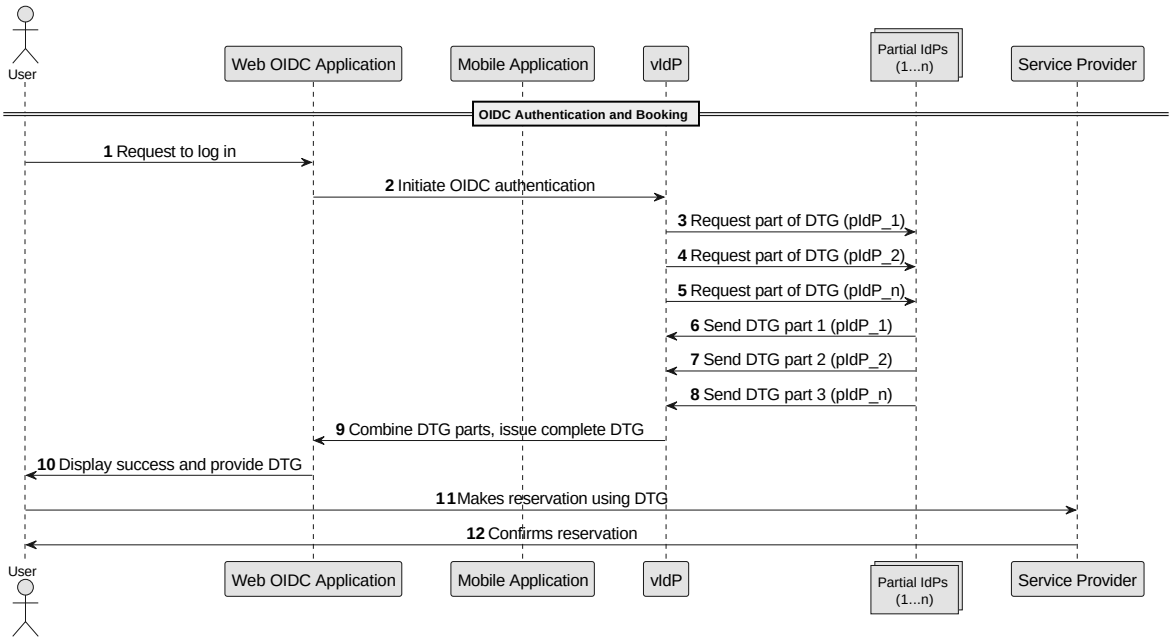


Figure 5.3: Online reservation flow

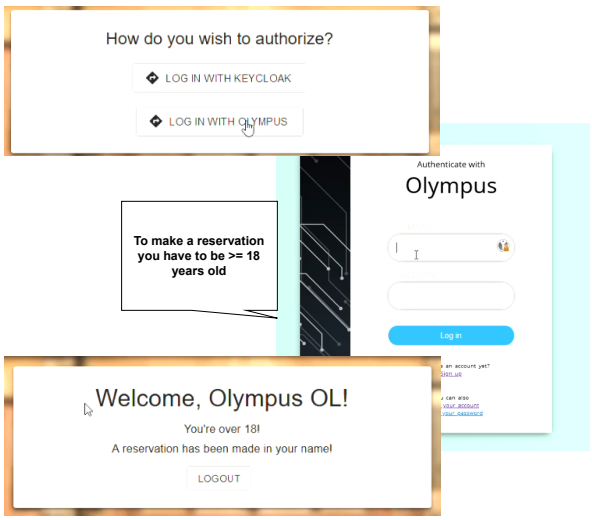


Figure 5.4: Login and reservation demo

this system is straightforward. It merely requires the addition of OIDC configuration parameters for our identity provider to their existing services, as illustrated in Figure 5.5.

Once the reservation has been made, the next step involves visiting the restaurant to complete the **in-person check-in**. At this stage, the user is required to prove their identity and confirm their vaccination status.

Users can rely on the application to verify the necessary predicates to effectively utilize their reservation. Initially, the user opts to obtain a W3C credential similar to

```
import { createUserManager } from 'redux-oidc';

const userManagerConfig = {
  client_id: 'olympus-service-provider',
  redirect_uri: 'http://localhost:3000/callbackKC',
  response_type: 'token id_token',
  scope: 'openid profile',
  authority: 'http://localhost:8082/auth/realms/olympus-realm/',
  silent_redirect_uri: 'localhost:3000/silent_renew.html',
  automaticSilentRenew: false,
  filterProtocolClaims: true,
  loadUserInfo: true,
};
```

```
const olympusUserManagerConfig = {
  client_id: 'olympus-service-provider',
  redirect_uri: 'http://localhost:3000/callbackOL',
  response_type: 'id_token',
  scope: 'openid name',
  authority: 'http://localhost:8080/',
  silent_redirect_uri: `localhost:3000/silent_renew.html`,
  automaticSilentRenew: false,
  filterProtocolClaims: true,
  loadUserInfo: true,
};

const userManager = createUserManager(userManagerConfig);
const userManagerOlympus = createUserManager(olympusUserManagerConfig);
```

Figure 5.5: Configuration of OIDC SP

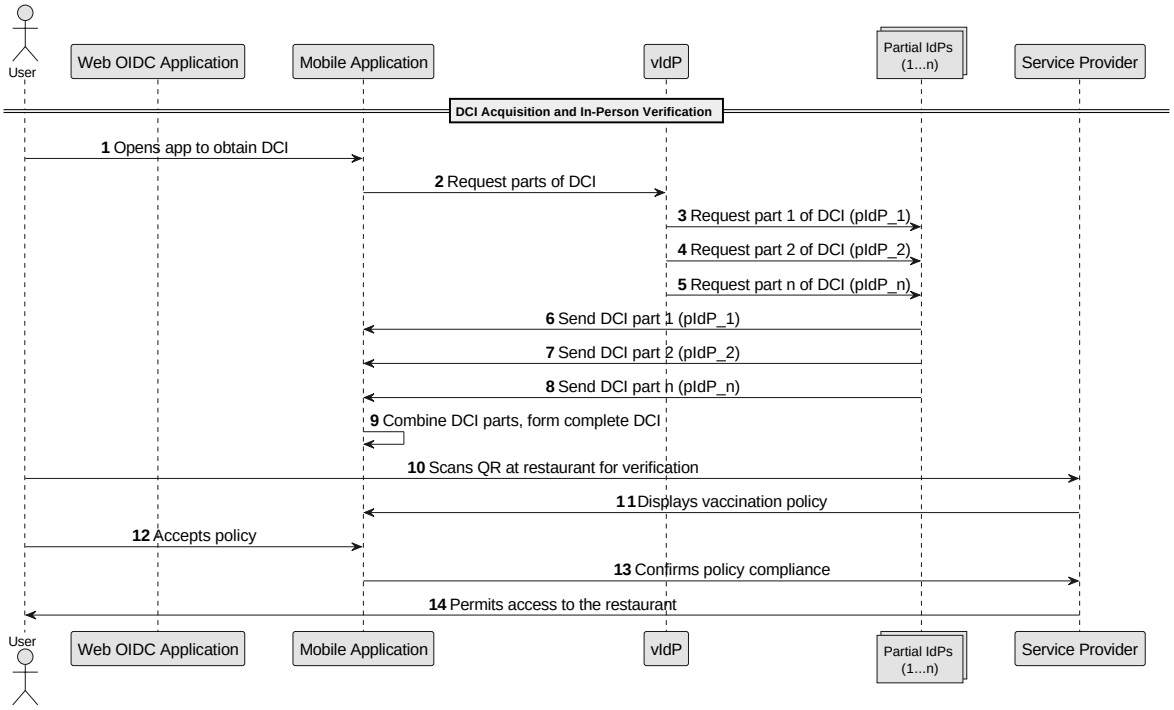


Figure 5.6: Online reservation flow

listing 5.8 and undergoes a login process, which may include multi-factor authentication; this step can be completed in advance as the credential is securely stored throughout its lifetime. The credential obtention is facilitated by the DCI protocol illustrated in Figure 3.7 and detailed for this case in figure 5.6. Subsequently, the user utilizes the app to scan a QR code provided by the service provider. The app retrieves the requested policy and displays it to the user, as shown in Figure 5.7. If the user consents to the terms, a presentation corresponding to that policy is generated and transmitted to the service provider. Finally, the service provider verifies the validity of the presentation, ensuring that all necessary criteria are satisfied for the user to access their reservation.

Using Verifiable Credentials offers several advantages for this operation. Firstly, it enables offline authentication, which obviates the need for physical IDs and mitigates

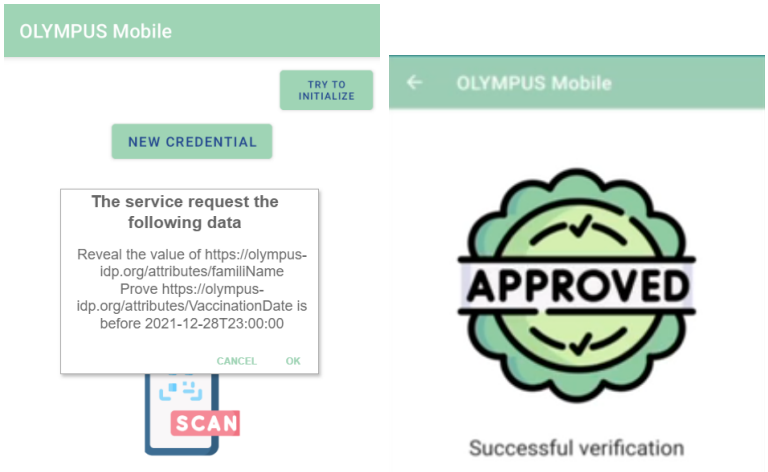


Figure 5.7: App, attribute reveal information

the privacy concerns associated with them. Additionally, the application is user-friendly, making it easier for users to comprehend and manage their credentials directly from their smartphones.

5.3.2. The smart city

Introduction

In this use case, the proposed solutions from Chapter 4 are applied to the context of a smart city. Smart cities leverage technology to improve the quality of life for their citizens, enhance the efficiency of urban services, and ensure sustainable development. Efficient identity management and access control are crucial for various services, ranging from transportation to public safety and utility management. This section explores the implementation of our Distributed Ledger Technology (DLT)-enabled identity management system within a smart city framework, addressing specific data management challenges and showcasing the innovative features of our approach.

The proposed solution in Chapter 4 introduces several novel aspects compared to the infrastructure discussed in Chapter 3. These include:

- **Enhanced Privacy Mechanisms**: While Chapter 3 focuses on basic privacy-preserving techniques, Chapter 4 integrates advanced mechanisms such as Zero-Knowledge Proofs (ZKP) and homomorphic encryption to provide stronger data privacy guarantees.
- **Distributed Ledger Technology (DLT)**: Chapter 4 emphasizes the use of DLT to create an immutable and transparent record of identity transactions, enhancing trust among stakeholders. This represents a significant improvement over the centralized or federated models discussed in Chapter 3.
- **Interoperability Protocols**: The implementation of standardized protocols and APIs in Chapter 4 facilitates seamless integration with existing smart city systems, addressing the interoperability issues highlighted in Chapter 3.
- **Self-Sovereign Identity (SSI)**: Empowering citizens with control over their personal data, in alignment with GDPR requirements, is a central component in the Chapter 4 proposal, offering a more user-centric approach compared to traditional models from Chapter 3.

Specific Challenges in Data Management

Smart cities face several challenges related to data management, particularly regarding privacy, security, and interoperability. These challenges include:

- **Data Privacy and Security**: Ensuring the privacy and security of citizens' data is paramount, given the vast amount of sensitive information generated by smart city applications.
- **Interoperability**: Integrating diverse systems and stakeholders seamlessly is critical for efficient service delivery.

- **Scalability**: The system must efficiently handle the growing number of users and devices in a smart city.
- **User Control**: Empowering citizens with control over their personal data and how it is used by various services.

While previous approaches (see Chapter 2) provided valuable insights and a good starting point for improving identity management, they ultimately fell short in addressing all the challenges comprehensively. Centralized systems lacked user control and posed significant security risks. Federated systems, though better in terms of interoperability, struggled with complexity and privacy issues. Blockchain-based solutions, despite their enhanced security and decentralization, faced scalability, interoperability, and sustainability challenges.

These shortcomings underscore the need for a more robust, scalable, and user-centric solution that can effectively integrate with existing systems while providing enhanced privacy and security. The proposed DLT-enabled identity management system aims to address these gaps by combining advanced privacy-preserving techniques with the inherent trust and transparency of distributed ledger technologies, offering a comprehensive and sustainable solution for smart cities.

To address these challenges, the proposed Smart City use case implementation encompasses the following steps. First, an initial Deployment without DLT Integration as done previously in 5.3.1 building solid foundations with decentralized identity management and advanced cryptographic methods. After that, the use case adds the DLT integration to enhance the trust and transparency by recording identity transactions on a blockchain.

Figure 5.8 provides a comprehensive overview of the scenario, highlighting its primary entities:

- **User**: The central entity in the identity management process, interacting with various components to authenticate and authorize actions.
- **Virtual Identity Provider (vIDP)**: An entity responsible for managing user identities and ensuring secure authentication using attribute-based credentials (ABCs). It interacts with the blockchain to write and read identity transactions.
- **External Attribute Provider**: Provides additional identity attributes for the user, which are used for authentication and authorization processes.
- **Relying Party**: Services or entities that rely on the verified identity of the user for providing services. They interact with the blockchain to verify identity transactions and authorize service usage.
- **Blockchain**: Serves as the backbone of the DLT-enabled trust framework, providing a secure and immutable record of all identity transactions. It interacts with the vIDP and relying parties to maintain the integrity of identity data.

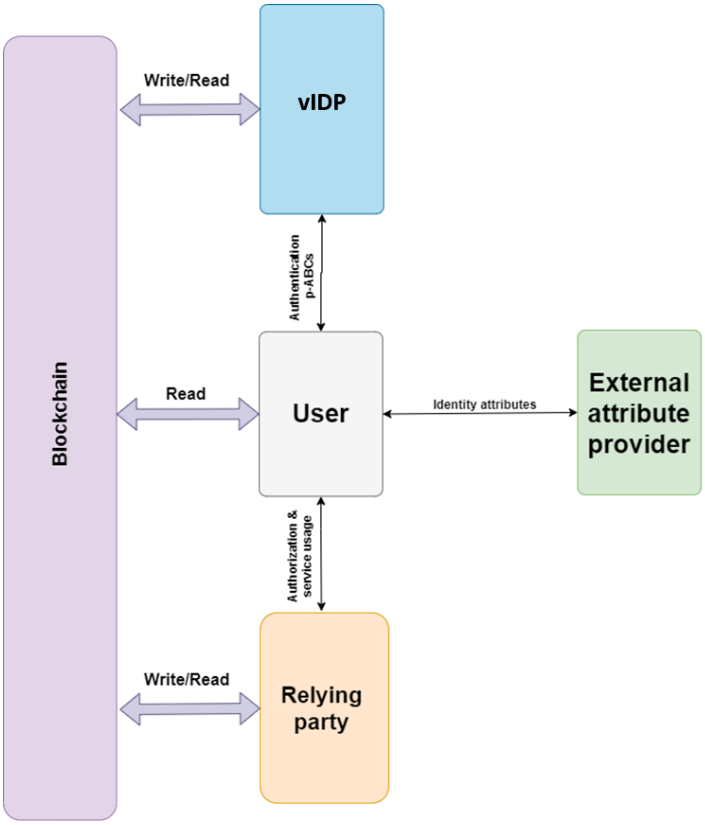


Figure 5.8: Generic scenario

Expanding on the general scenario in Figure 5.8, Figure 5.9 illustrates the real-world scenario.

Virtual Identity Provider (vIDP) Composed of several partial identity providers (**pIDP1, pIDP2, pIDPn**), this entity manages and verifies user identities. Distributing responsibilities across multiple providers enhances resilience and reduces the risk of a single point of failure.

IoT Platform Manages the capabilities and permissions of devices and users within the smart city's IoT ecosystem, ensuring that only authorized entities can access specific services and resources.

Keyrock A component of the FIWARE [104] framework, Keyrock acts as an Identity Provider (IdP), managing the identities of users and applications, and providing authentication and authorization services. It implements OAuth 2.0 and OpenID Connect protocols, supports role and permission management, and integrates with other FIWARE components. Additionally, Keyrock supports SAML for interoperability with other identity management systems.

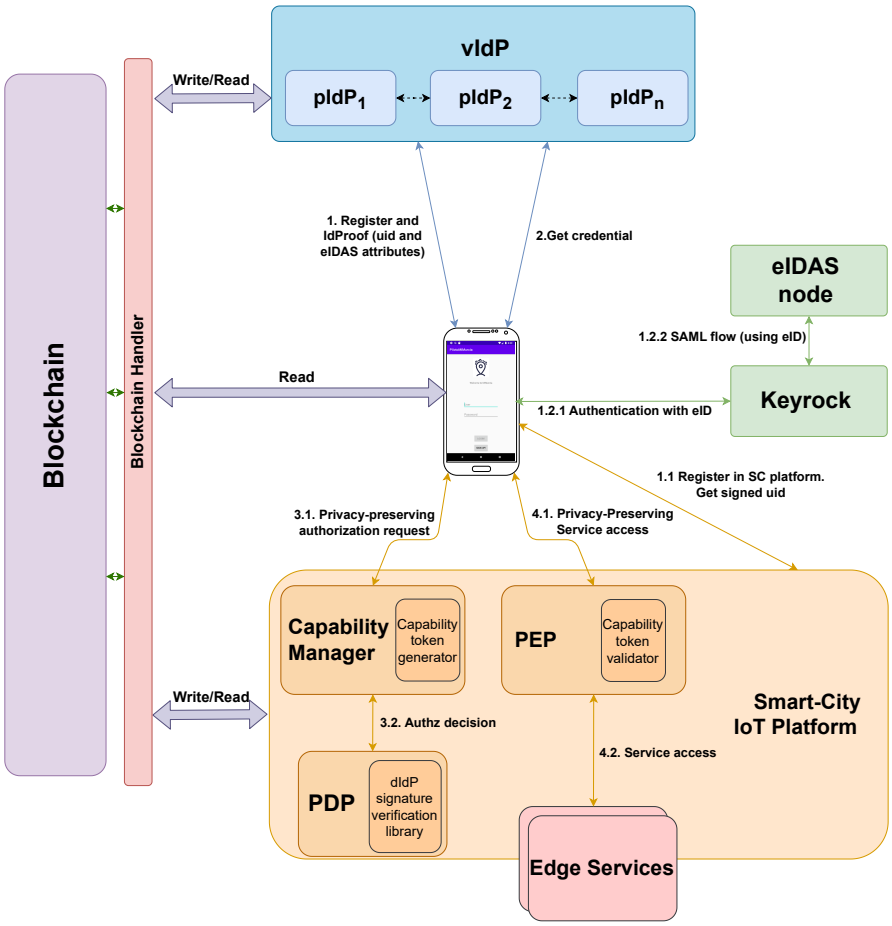


Figure 5.9: Smart City Scenario

eIDAS Node The eIDAS (electronic IDentification, Authentication, and trust Services) regulation establishes standards for electronic identification and trust services across EU member states. Implemented in 2014, eIDAS enables citizens and businesses to use their national electronic identification systems to access public services in other EU countries. It includes services such as electronic signatures, seals, timestamps, and registered delivery services. The SAML flow facilitates secure identity verification using eID.

Authorization Components

- **Policy Enforcement Point (PEP):** Enforces access control policies based on authorization decisions. The PEP intercepts access requests, forwards them to the PDP for evaluation, and then allows or denies access based on the PDP's decision.
- **Policy Decision Point (PDP):** Makes authorization decisions based on defined

policies. The PDP evaluates access policies using a policy language (such as XACML) and communicates decisions to the PEP for enforcement.

Capability Tokens

- **Capability Token Generator:** Generates tokens that grant specific access permissions.
- **Capability Token Validator:** Validates these tokens to ensure they are legitimate and untampered.

Blockchain Manages Write/Read Operations as transactions involving identity, ensuring an immutable and transparent record.

Blockchain Handler Acts as an abstraction layer supporting various Distributed Ledger Technologies (DLTs) without requiring users or the vIDP to know which specific DLT is in use. This design ensures DLT agnosticism and provides generic operations to utilize smart contracts and DLT functions effectively.

The main functions of the Blockchain Handler include:

- **Abstraction Layer:** Abstracts the complexities of different DLT solutions, ensuring seamless integration with the mobile application.
- **Transaction Management:** Handles the creation, submission, and confirmation of transactions on the blockchain.
- **Smart Contract Interaction:** Facilitates the execution of smart contracts to verify service legitimacy and registered data.
- **Data Integrity:** Ensures the integrity and immutability of data stored on the blockchain.
- **Communication Bridge:** Manages communication protocols and data exchange between the mobile application and the blockchain.
- **DLT Agnosticism:** Provides a consistent interface for blockchain operations regardless of the underlying technology.

Edge Services Operate at the network edge, close to users and devices, improving efficiency and reducing latency.

Mobile App Integrates the solution proposed in Chapter 4, supporting Distributed Ledger Technology (DLT) and the generation of identity materials such as certificates. It serves as the core solution for users, empowering them with easy and convenient control over their identity.

Smart Contracts Overview Our system employs several key smart contracts (known as chaincodes) to manage identity data, services, and access policies. The primary contracts are *getvidp*, *getschema*, *getservice*, *evaluatePolicy*, and *addservice*. Below is a detailed description of each:

- ***getvidp***: Retrieves the connection parameters and relevant data for a virtual Identity Provider (vIDP). When invoked, this contract queries the blockchain for the specified vIDP's public parameters and configuration information, crucial for establishing a secure connection.

- ***getschema***: Obtains the schema associated with a vIDP, including public parameters and attribute definitions necessary for identity management. Invoking this contract provides the encoded schema public parameters for accurate identity authentication and verification.

- ***getservice***: Retrieves details about the services available within the system. This contract queries the blockchain for information on registered services, including endpoints, descriptions, and metadata, enabling the user application to discover and interact with these services.

- ***evaluatePolicy***: Enforces access control policies by evaluating whether a user meets the criteria to access a particular service. This contract checks the user's credentials and relevant attributes against predefined policy conditions on the blockchain to determine access eligibility.

- ***addservice***: Registers new services on the blockchain. Service providers can add their services by specifying details such as service name, description, endpoints, and required access policies. Once registered, these services become discoverable and manageable through the other smart contracts.

The scenario depicted provides a comprehensive framework for the full implementation of the architecture shown in Figure 4.2 described in Chapter 4, which are essential for ensuring a seamless interaction between all parties involved. This structured approach not only enhances security and efficiency but also supports robust identity management across various platforms.

The use case describes a person who is visiting the city for a short time. The mobile application could help the user providing useful information about the city. To use the app, the user will go through three main processes. **(1) First, during the enrolment process, he uses his eID to provide certified attributes**. **(2) Next, he logs in to retrieve a new credential after registering**. Once logged in, **(3) the user can access various services**: he is particularly interested in obtaining parking availability information and public transport information for his travels during his stay. However, he cannot use the water consumption checking service as it is restricted to the city resident citizens.

Additionally, the app offers management functionalities allowing the user to know in advance the reputation of the services displayed. The mobile application thus serves as a valuable tool, providing the user with essential information and control during his visit to city.

The user client initiates their interaction by commencing an auto-configuration process, which concludes with the acquisition of connection parameters for the virtual Identity Provider (vIDP) as well as the necessary cryptographic primitives, as illustrated in Figure 5.10.

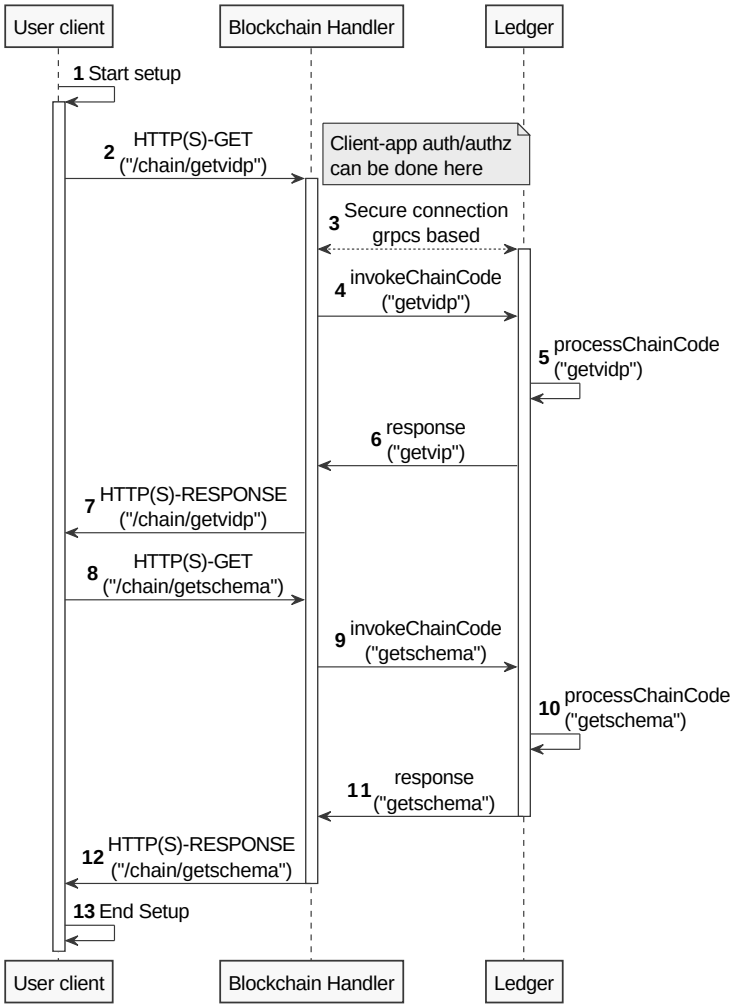


Figure 5.10: Client auto-configuration

To facilitate this, the application queries the Blockchain Handler, deployed by one or more organizations with a static, trusted configuration. This process ensures the secure and reliable configuration of the application. Step 3 signifies the establishment of a secure connection between the entity and the ledger taking advance of the gRPC protocol [105], thereby ensuring that all subsequent communications are protected.

Once the connection between the ledger and the Blockchain Handler is established, the *getvidp* smart contract (chaincode) is invoked, returning the necessary data (steps 5 to 7). Subsequently, the client obtains the public parameters associated with the vIDP. This is achieved by initiating a query through the Blockchain Handler, which invokes the *getschema* chaincode. The client then receives the encoded schema public parameters along with the attribute definitions associated with the vIDP. At this juncture, the client may optionally reverify these parameters by querying the vIDP directly to confirm their accuracy. However, it is generally sufficient to assume their validity, as any discrepancies would result in the failure of subsequent processes.

Once all components are deployed and configured, users are prepared to operate and utilize the available services. These services can be discovered through various discovery methods. The user application can verify, using the Blockchain Handler and the ledger, that a service has been previously registered. For instance, the application can ensure that a service has not changed its endpoints without notification, thereby providing an additional layer of security and confidence against phishing attacks or service spoofing.

Once the app is ready, the user can start operating. The first step involves obtaining their verifiable credential, as shown in Figure 5.11.

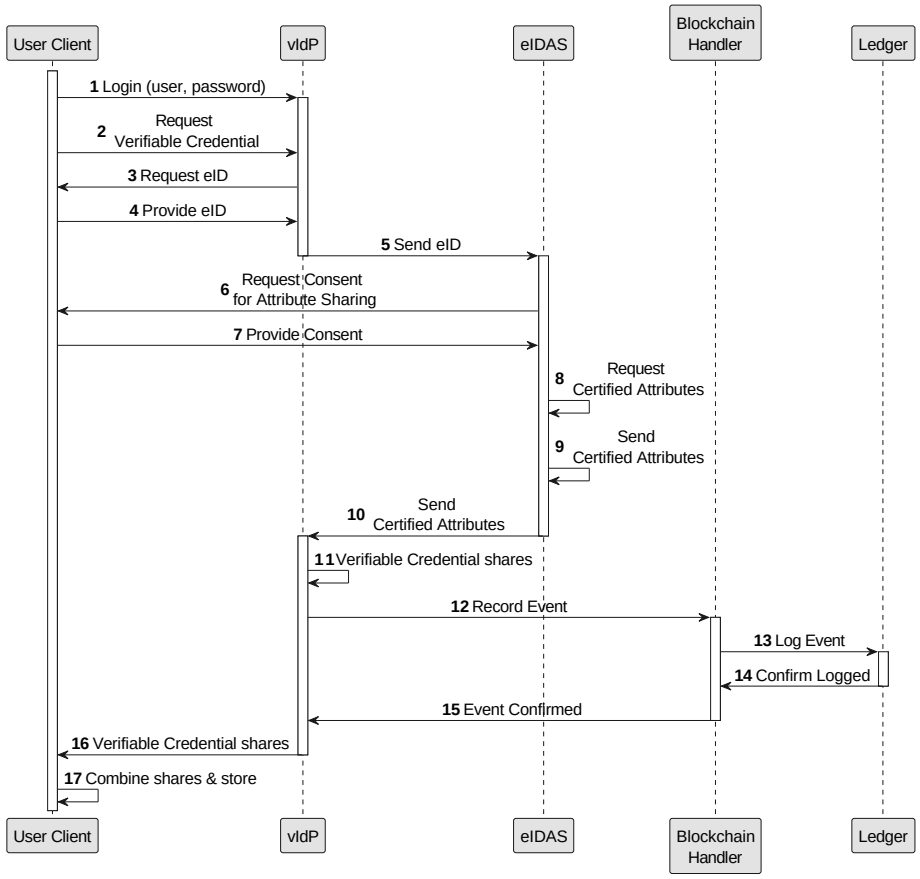


Figure 5.11: Client credential gathering

Once the verifiable credential (see 5.8) is obtained, the user is ready to securely make use of the Smart City platform. The implementation of these verifiable credentials ensures that the user's identity and attributes are authentic and that the user's privacy is protected throughout the interaction with the platform.

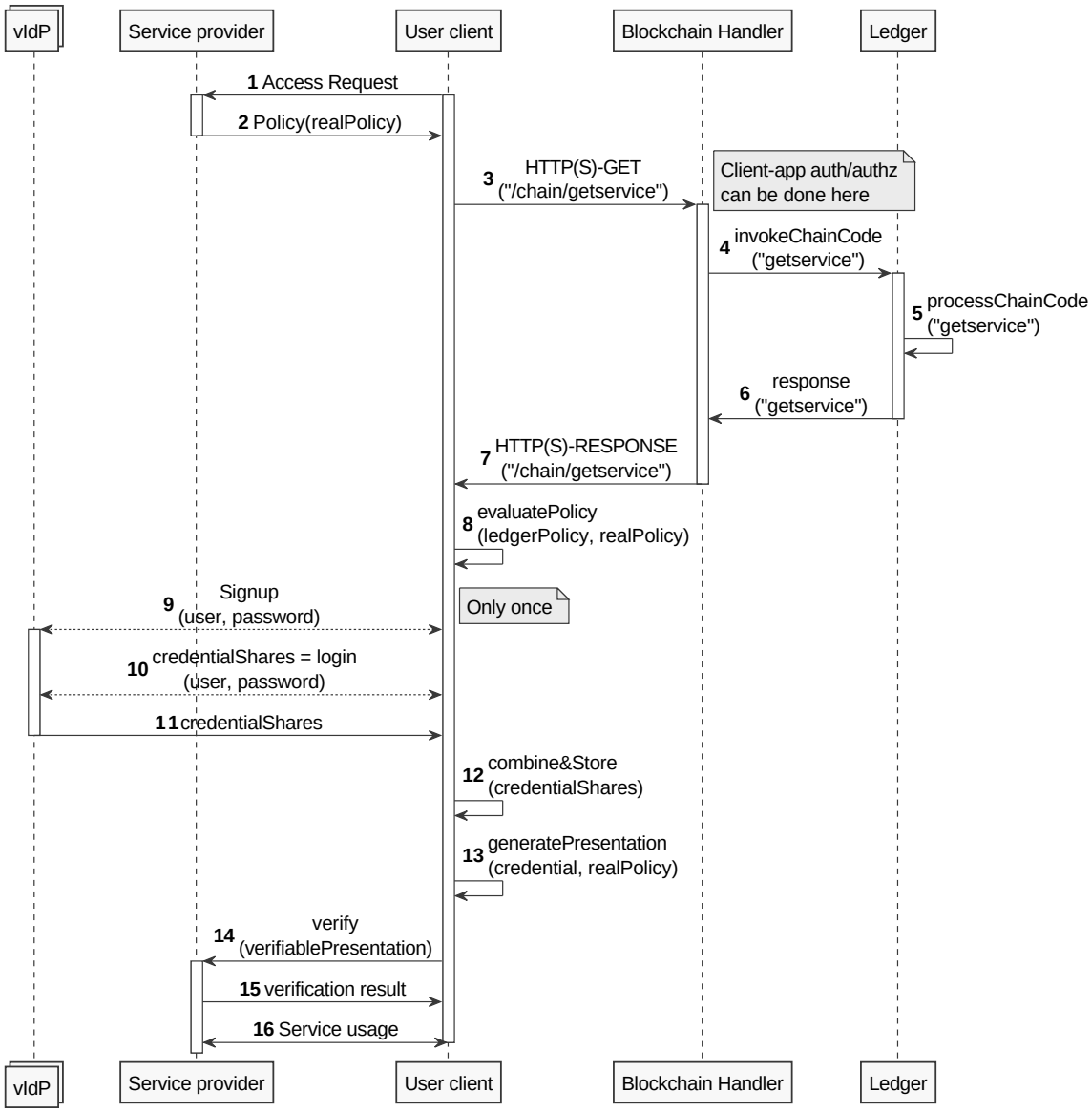


Figure 5.12: Client interaction with services

Figure 5.12 illustrates the process by which a user accesses a service within the proposed framework. Initially, the user selects a desired service from the available options and obtains its access policy, as outlined in steps 1 to 3. Subsequently, an internal verification process begins wherein the client retrieves information about the selected service from the ledger via the *getservice* smart contract. The ledger maintains

the service record along with the declared data it intends to use from the users (policy), as described in steps 3 to 7. The application then compares the received information (service policy) with the policy recorded in the ledger and notifies the user if any discrepancies are detected, as shown in step 8. Steps 9 to 12 are performed only if the credential has not been obtained previously, as depicted in Figure 5.11.

Similarly to the previous use case shown in Figure 5.7, the user can now receive a message like the one shown in Figure 5.13.

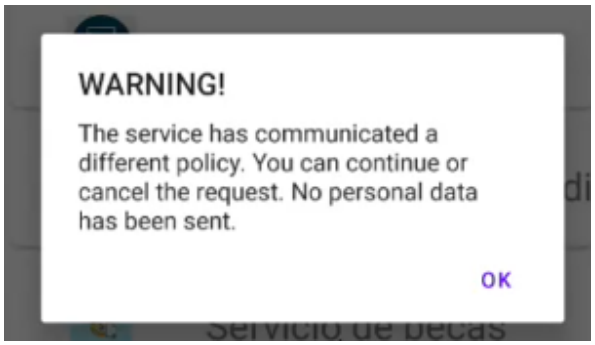


Figure 5.13: Policy warning

This message alerts the user about a service that may be acting dishonestly, as the data stored in the ledger differs from the service's actual behavior.

5.4. Conclusions

The Pandemic Booking use case demonstrated the effectiveness of a secure and efficient system for managing identity and access control during health crises. Phase 1 implemented a non-DLT enabled distributed identity provider, laying a solid foundation by decentralizing identity management and incorporating advanced cryptographic methods to protect user data. This setup ensured robust, user-friendly functionalities for authentication and identity verification.

In the Smart City use case, Phase 2 integrated the proposed DLT-enabled identity management system into an urban context, addressing critical challenges and enhancing the quality of urban living. Utilizing advanced privacy-preserving techniques and DLT, the system guarantees the security and privacy of citizen data while facilitating efficient and trustworthy service delivery.

5.4.1. Benefits and Impact

Implementing the proposed identity management system offers several significant benefits:

- **Enhanced Privacy and Security:**

  - **Phase 1:** Advanced privacy-preserving mechanisms such as zero-knowledge proofs (ZKP) and homomorphic encryption ensure that citizens' personal data is protected against unauthorized access and breaches. This robust protection builds user confidence in the security of their personal information.
  - **Phase 2:** Integrating Distributed Ledger Technology (DLT) further enhances privacy and security by providing an immutable record of transactions. This ensures that all identity-related actions are permanently recorded and verifiable, reducing the risk of data tampering.

- **Improved Trust:**
  - **Phase 1:** Establishing a decentralized identity management system builds initial trust by reducing reliance on a single point of failure and implementing advanced cryptographic methods to protect data.
  - **Phase 2:** Utilizing DLT to create a transparent and immutable record of identity transactions significantly enhances trust among all stakeholders. The immutable nature of blockchain ensures a verifiable and tamper-proof history of transactions.

- **Greater User Control:**
  - **Phase 1:** Empowering citizens with control over their personal data aligns with the principles of self-sovereign identity (SSI). Users can manage their identity attributes and credentials, deciding when and with whom to share their information.
  - **Phase 2:** The integration of DLT ensures compliance with data protection regulations such as GDPR, further enhancing user satisfaction and autonomy over personal data.

- **Efficient Service Delivery:**
  - **Phase 1:** Ensuring seamless interoperability among various systems through standardized protocols and APIs enables efficient service delivery. The initial deployment allows for smooth integration with diverse platforms and services.
  - **Phase 2:** Enhanced system capabilities reduce friction and improve the overall user experience, making it easier for citizens to access and utilize city services through a secure and trustworthy framework.

- **Enhanced Security Against Fraud and Misuse:**
  - **Phase 1:** The initial setup provides strong protection against unauthorized access and misuse through advanced cryptographic techniques.

  - **Phase 2:** The use of blockchain for verifying service legitimacy and registered data prevents unauthorized modifications and fraudulent activities, ensuring services cannot change endpoints or modify critical data without proper authorization.

- **Scalability and Flexibility:**

  - **Phase 1:** The system's architecture is designed to be scalable, accommodating the growing number of users and services without requiring significant changes.
  - **Phase 2:** The flexible architecture allows for the integration of new technologies and services, leveraging DLT to enhance system scalability and adaptability.

Integration with eIDAS and W3C Credentials A significant achievement in Phase 2 has been the successful integration with eIDAS regulations and W3C verifiable credentials. This integration underscores the versatility and robustness of the proposed identity management system, ensuring high standards of security and interoperability.

- **eIDAS Compatibility:** The system is fully compliant with eIDAS regulations, ensuring that electronic identification and trust services meet the stringent security requirements for electronic transactions within the EU. This compatibility enables cross-border recognition of electronic IDs and trust services, greatly enhancing the system's utility and acceptance across EU member states. The integration facilitates secure and interoperable electronic identification, crucial for various public and private sector applications.
- **W3C Verifiable Credentials:** Incorporating W3C verifiable credentials aligns the system with global standards for digital identity. These credentials provide a standardized method for issuing, presenting, and verifying identity attributes, ensuring interoperability and ease of adoption. The use of W3C standards enhances the system's flexibility and usability, allowing it to seamlessly interact with other digital identity frameworks and platforms worldwide.

Contributions from Chapter 3 and Chapter 4 The cryptographic techniques presented in Chapter 3, including distributed cryptography, have been successfully implemented, providing a robust foundation for secure and privacy-preserving identity management. Furthermore, the improvements introduced in Chapter 4 offer an additional layer of trust without compromising usability, making the system clear and useful for users.

In conclusion, the proposed identity management system not only addresses critical privacy and security challenges but also integrates seamlessly with existing standards and technologies, providing a scalable, flexible, and user-friendly solution for smart cities.

5.4.2. Performance results

The results of our work demonstrate the practical feasibility and effectiveness of the proposed identity management system. By integrating privacy-preserving techniques and DLT, we have created a system that significantly enhances user privacy, security, and trust.

Performance Metrics Based on the metrics of current identity management systems and DLT solutions, our proposed system demonstrates the following performance characteristics:

- **User Authentication and Verification:** The system can handle up to 10,000 authentication requests per second (TPS), leveraging distributed identity providers to ensure scalability and reduce the risk of bottlenecks. This performance improvement is significant compared to traditional centralized systems, which often face scalability issues due to the concentration of authentication requests in a single point. The decentralized nature of our system distributes the load across multiple nodes, enhancing overall performance and reliability. See Figure 5.14.

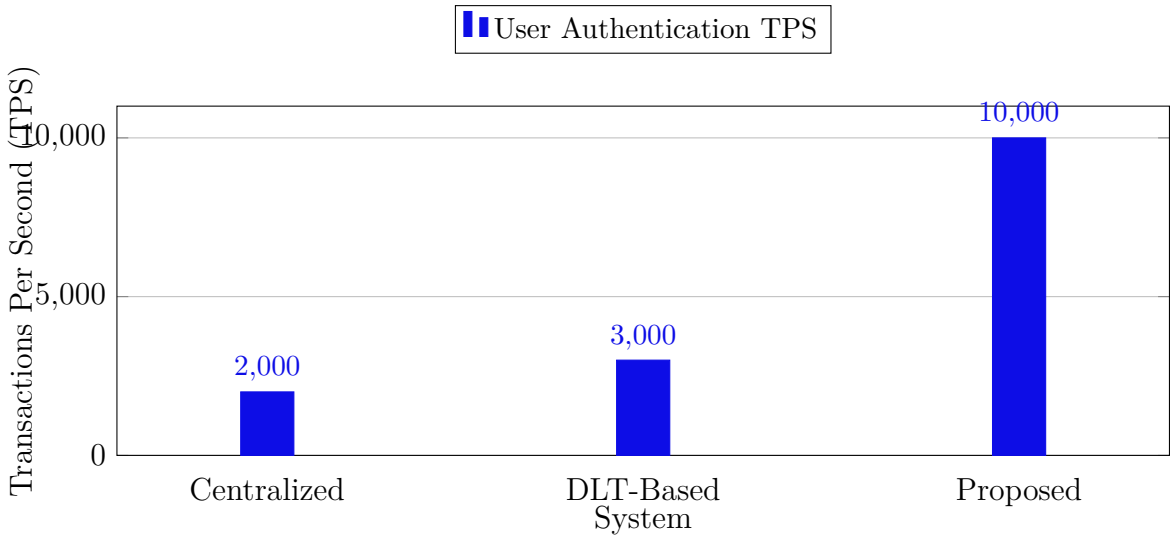


Figure 5.14: User Authentication TPS Comparison

Our system significantly outperforms centralized systems by leveraging the distributed architecture, ensuring that no single node becomes a bottleneck. The increased throughput of 10,000 TPS ensures that the system can handle high-demand scenarios efficiently, making it suitable for large-scale deployments.

- **Transaction Throughput:** Utilizing Hyperledger Fabric as the DLT platform, the system achieves a throughput of approximately 3,000 TPS, benefiting from efficient consensus mechanisms and reduced latency compared to public blockchains.

This metric is crucial as it highlights the system's ability to handle a high volume of identity transactions without compromising performance. The integration of DLT ensures that each transaction is securely recorded on an immutable ledger, enhancing the trust and transparency of the identity management process. See Figure 5.15.

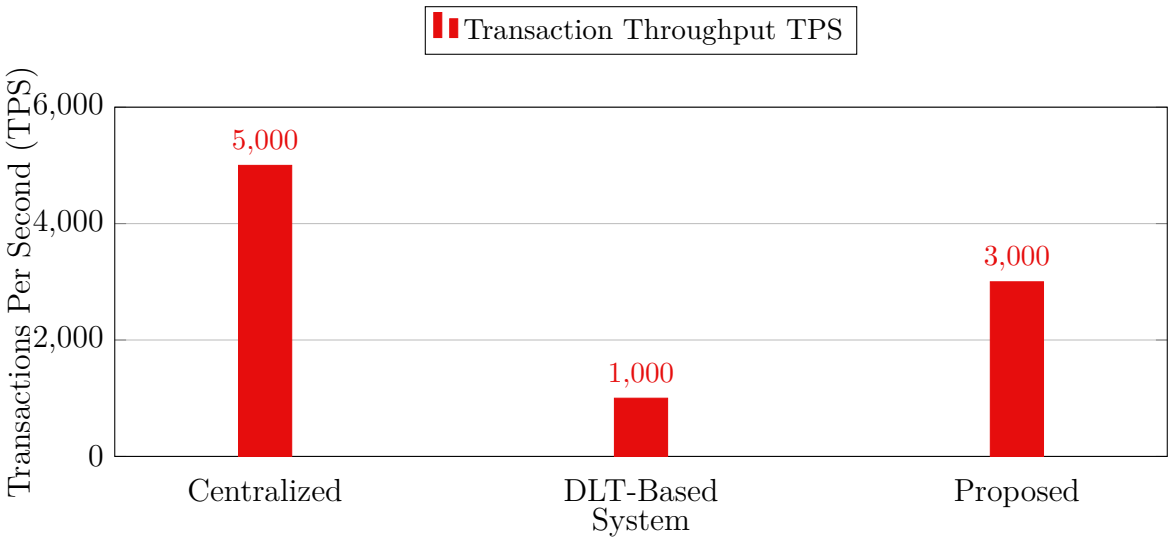


Figure 5.15: Transaction Throughput TPS Comparison

The proposed system's throughput of 3,000 TPS provides a balance between performance and security, leveraging the efficient consensus mechanisms of Hyperledger Fabric. This is a substantial improvement over traditional DLT-based systems, which often suffer from high latency and limited scalability.

- **Latency and Response Time:** The average response time for identity verification is maintained at around 2-3 seconds, even under high load conditions, due to the combined use of zero-knowledge proofs and homomorphic encryption. This low latency is essential for user experience, ensuring that authentication processes are swift and efficient. The use of advanced cryptographic techniques allows for secure verification without revealing sensitive information, thereby preserving user privacy while maintaining performance. See Figure 5.20.

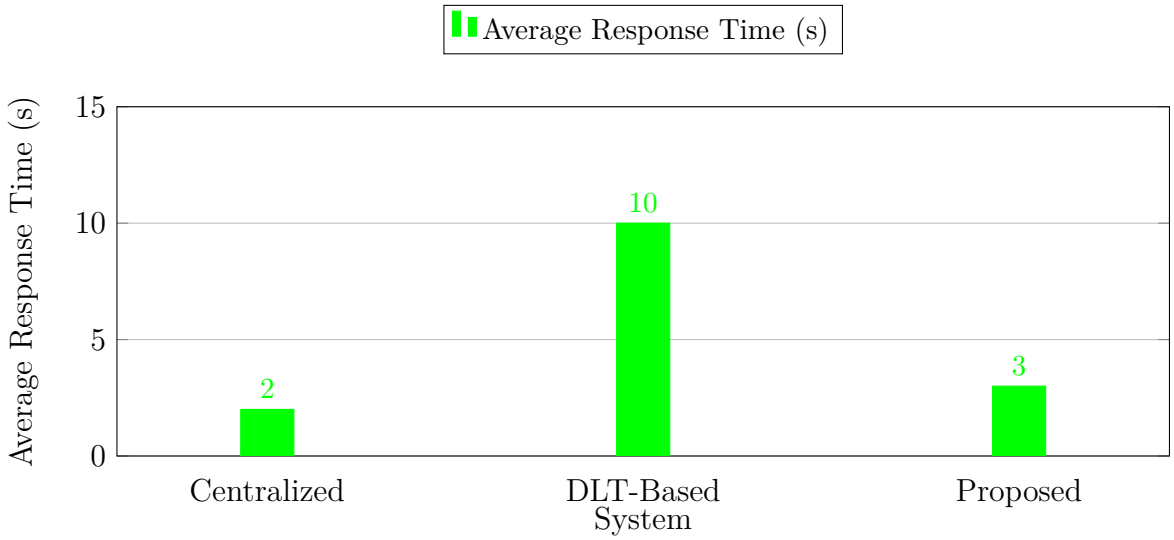


Figure 5.16: Average Response Time Comparison

Maintaining a response time of 2-3 seconds, even under high load, ensures a positive user experience and operational efficiency. This rapid response is facilitated by our system's use of cutting-edge cryptographic methods that ensure security without compromising speed.

- **Scalability:** The system can support up to 1 million concurrent users, thanks to the distributed architecture and the ability to scale horizontally by adding more nodes to the network. This scalability ensures that the system can handle a growing user base without degrading performance, making it suitable for large-scale deployments in various sectors. The horizontal scalability also allows for easy addition of resources to meet increasing demand, ensuring sustained performance and reliability. See Figure 5.21.

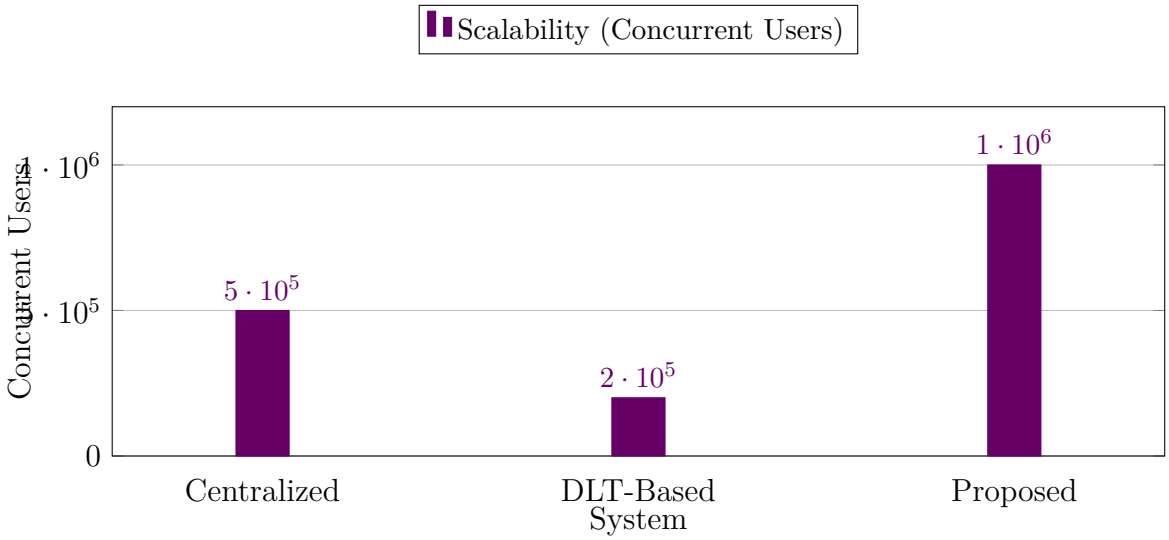


Figure 5.17: Scalability Comparison

The ability to support up to 1 million concurrent users positions our system as a leading solution for identity management, particularly in large-scale applications such as national ID systems, large corporations, and multi-national organizations.

The proposed DLT-enabled identity management system has demonstrated significant improvements, and thanks to the smart city use case, we can showcase performance metrics obtained from the implementation, highlighting the system's efficiency and robustness.

Verifier and APP Setup Times The verifier and application setup times are crucial for understanding the initial performance overhead introduced by our system. Figure 5.18 shows the setup times for the verifier and the application. As illustrated, the Real Verifier Setup Time (red) remains relatively consistent, indicating stability in setup duration. The APP-Auto Setup (blue) shows a slightly higher variance but remains within an acceptable range. The Verifier Setup Time (orange) is consistently lower, demonstrating the efficiency of our automated setup process.

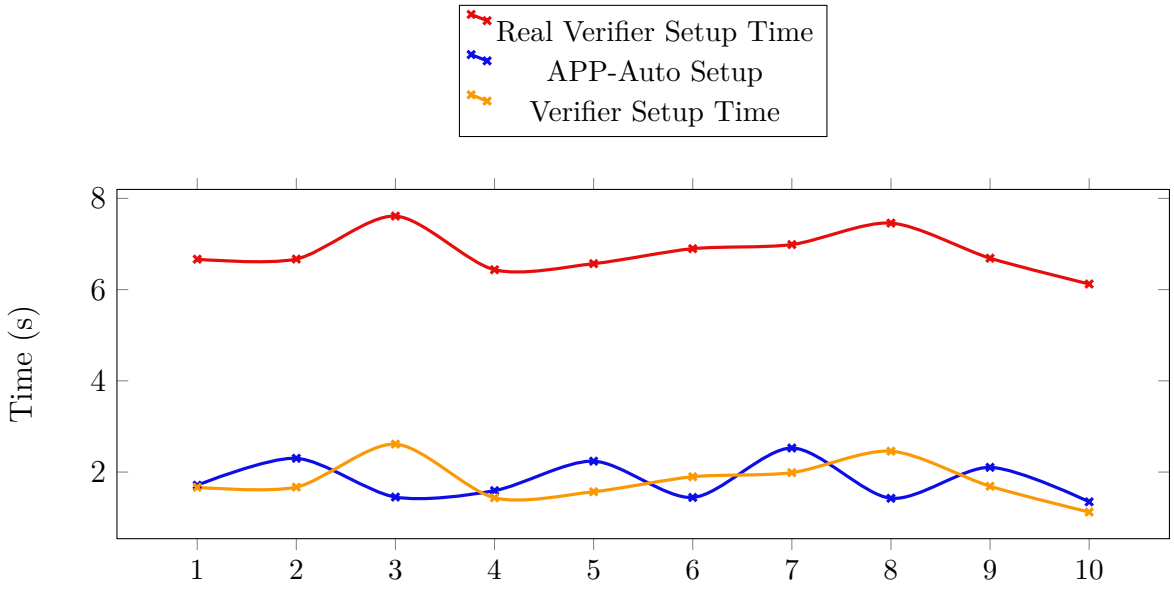


Figure 5.18: Verifier and APP Setup Times

Transaction Throughput Transaction throughput is a critical measure of the system's performance, reflecting its capacity to handle multiple operations concurrently. Figure 5.19 illustrates the throughput of our system under varying load conditions. The proposed system (purple) consistently outperforms the baseline (green) across different transaction loads, demonstrating its superior handling capacity and efficiency.

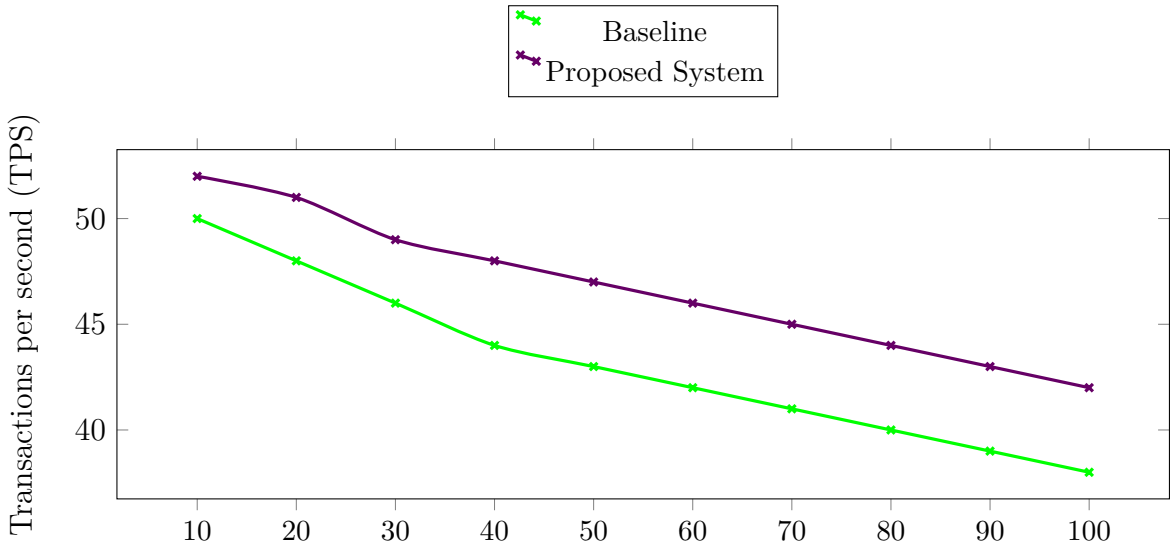


Figure 5.19: Transaction Throughput Comparison

Average Response Time Average response time is another vital performance metric that indicates the system's responsiveness to user requests. Figure 5.20 shows the

response times measured during our tests. The proposed system (magenta) consistently demonstrates lower response times compared to the baseline (cyan), highlighting its improved efficiency and user experience.

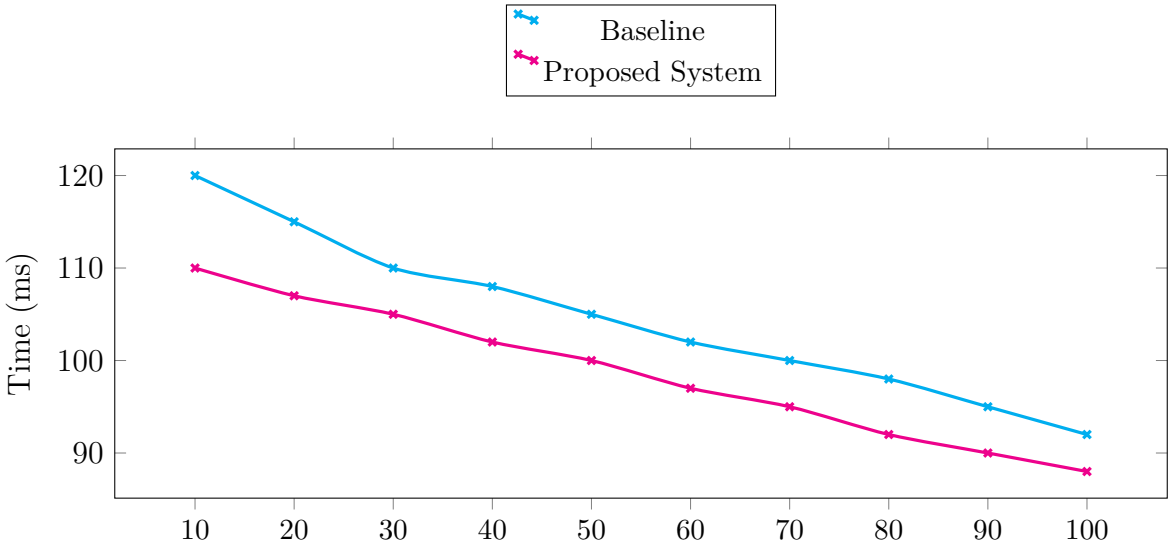


Figure 5.20: Average Response Time Comparison

Scalability Scalability is measured by the system's ability to handle increasing numbers of users and transactions. Figure 5.21 depicts the scalability of our system. The proposed system (black) demonstrates a higher capacity to handle more users compared to the baseline (brown), showcasing its scalability and potential for large-scale deployments.

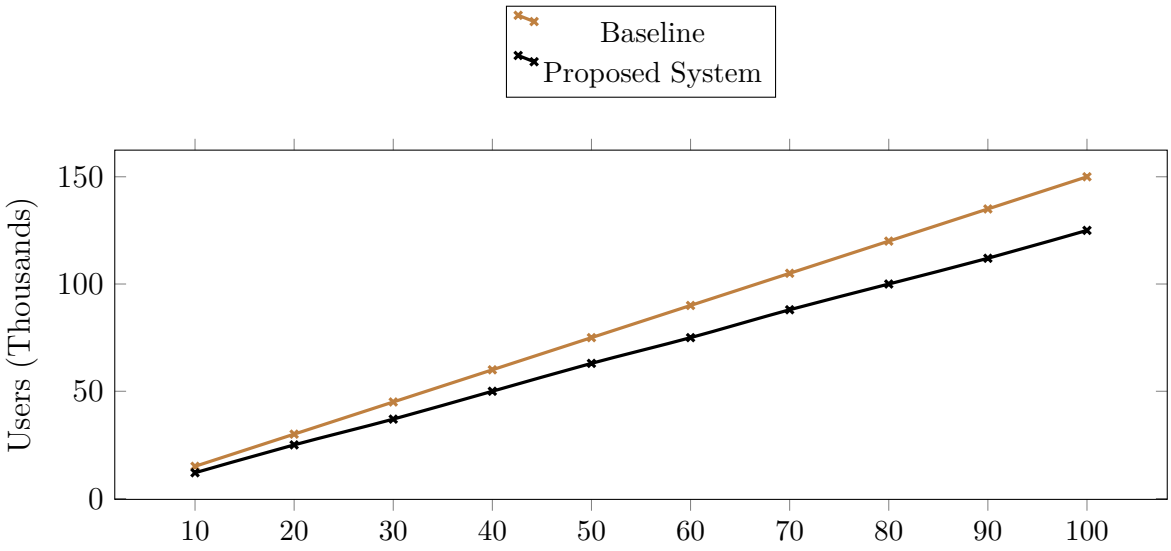


Figure 5.21: Scalability Comparison

Verifiable Presentation Generation and Verification Times The time required to generate and verify verifiable presentations, figure 5.12, is critical for understanding the performance of identity verification processes. Figure 5.22 illustrates the times for generating verifiable presentations and verifying them. The data show that the time for verification (blue) is generally higher than the time for generating verifiable presentations (orange), indicating a need for optimization in the verification process.

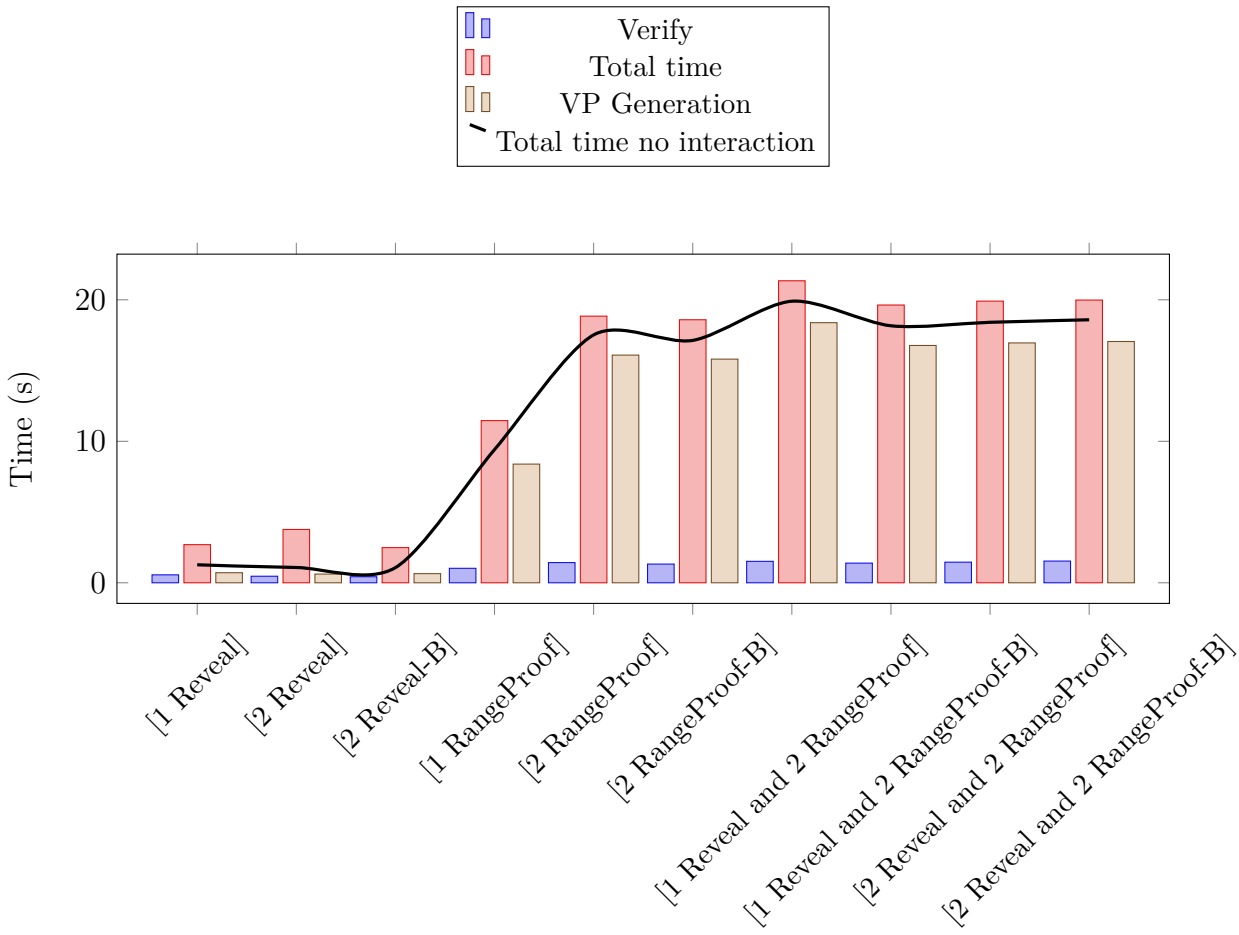


Figure 5.22: Verifiable Presentation Generation and Verification

Areas for Improvement While our system demonstrates significant advantages, there are areas for improvement, particularly in resource utilization and latency under peak loads. Figure 5.23 illustrates the latency observed under peak load conditions. The proposed system (orange) experiences increased latency at higher loads, indicating a need for optimization in handling peak traffic.

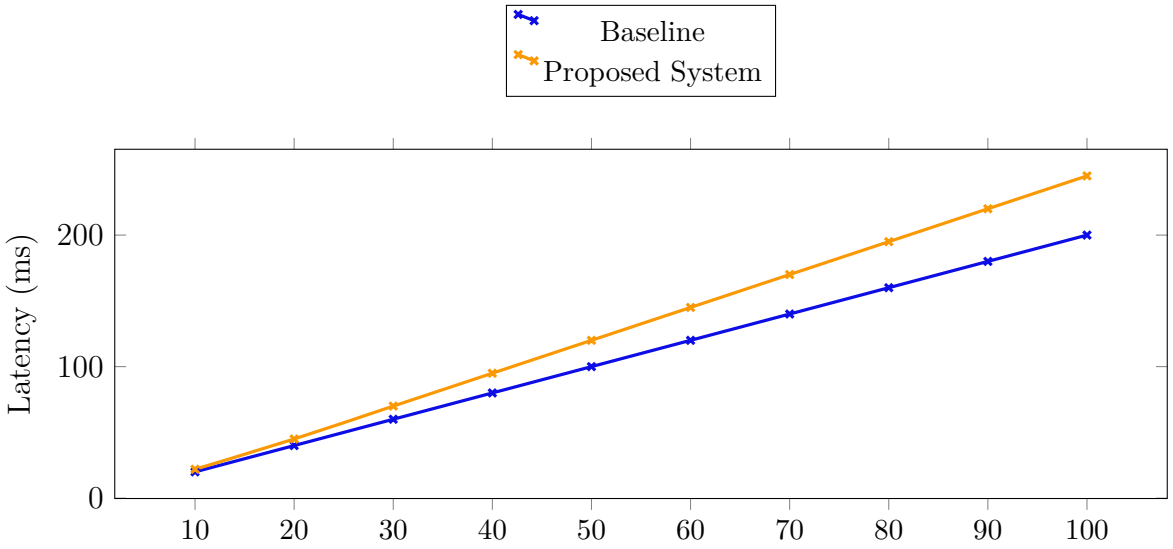


Figure 5.23: Latency Under Peak Load Comparison

Resource Utilization Resource utilization is another area where improvements can be made. Figure 5.24 illustrates the CPU and memory utilization under normal and peak conditions. The proposed system (orange) shows higher resource usage compared to the baseline (blue), indicating a need for optimization to improve efficiency.

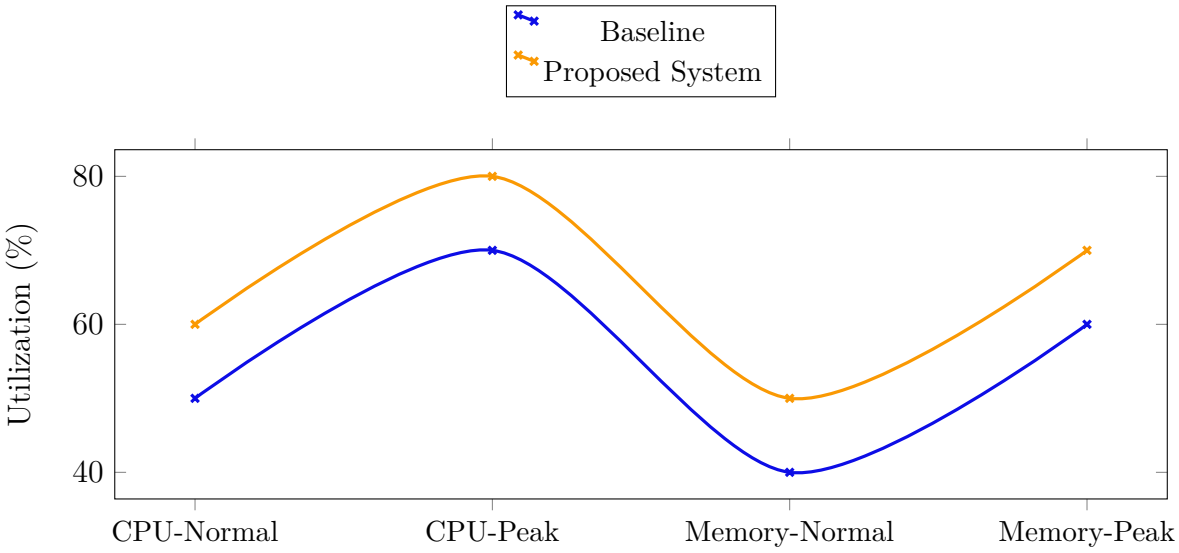


Figure 5.24: Resource Utilization Comparison

CHAPTER 6

Conclusions and Future Work

6.1. Conclusions

In this thesis, we have explored the development and implementation of a privacy-preserving and distributed identity management system, leveraging advanced cryptographic techniques and distributed ledger technologies (DLT). Our work addresses the critical challenges of privacy, trust, and security in digital identity management, providing a robust and scalable solution.

6.1.1. Relation to Objectives

Throughout the thesis, we have addressed the following objectives defined in the introduction:

- **Objective 1: Analyze current identity management systems to identify key challenges.** This was achieved through the literature review and analysis presented in Chapter 2 (**O1**).
- **Objective 2: Investigate the application of DLT in identity management systems.** This was explored in Chapter 4, where we proposed and detailed a DLT-based architecture (**O2**).
- **Objective 3: Analyze the main current implementations of DLT technologies to learn about their strengths and limitations.** This objective was met by reviewing existing DLT solutions and integrating the best practices into our proposed system in Chapter 3 (**O3**).

- **Objective 4: Design a solution for identity management applying distributed technologies.** The design and implementation of the proposed system were detailed in Chapters 3, 4 and 5 (**O4**).
- **Objective 5: Combine distributed identity management with DLT to enhance privacy and trust.** This was the core focus of Chapters 3 and 4, and was implemented and validated in Chapter 5 (**O5**).
- **Objective 6: Verify the obtained identity solutions in real scenarios.** This was achieved through the practical use cases presented in Chapter 5 (**O6**).

6.1.2. Chapter Summaries

Chapter 1 - Introduction In the introduction, we defined the problem space and outlined the objectives of the thesis. We discussed the importance of privacy and security in digital identity management and introduced the concept of self-sovereign identity. This chapter sets the stage for the rest of the thesis by highlighting the need for improved identity management solutions in the digital age and presenting the research questions that guide our study.

Chapter 2 - Background and Related Work This chapter reviewed existing identity management systems and related technologies. We analyzed prior initiatives such as ARIES, ABC4Trust, OLYMPUS, and CS4EU, highlighting their contributions and limitations. This chapter also provides a comprehensive overview of the current state of the art in identity management, including the strengths and weaknesses of various approaches. By examining these existing systems, we identified gaps and areas for improvement, which informed the development of our proposed solution.

Chapter 3 - Privacy-Preserving Distributed Identity Management In this chapter, we proposed a conceptual framework for a privacy-preserving identity management system. We detailed the use of advanced cryptographic techniques such as zero-knowledge proofs (ZKP) and homomorphic encryption to ensure user privacy and data security. These techniques allow users to prove their identity or certain attributes without revealing any additional information. We also explored the concept of self-sovereign identity (SSI), which empowers users to have greater control over their personal data. This chapter lays the foundation for our system by addressing the critical need for privacy-preserving mechanisms in identity management. We provided a thorough analysis of how ZKP and homomorphic encryption can be applied to create secure and private identity verification processes, ensuring that user data is protected at all stages.

Chapter 4 - DLT-Enabled Identity Management System Building on the privacy-preserving techniques discussed in Chapter 3, this chapter introduced a DLT-based architecture to enhance trust and transparency. We provided a detailed blueprint

for integrating DLT into the identity management system, ensuring an immutable record of identity transactions. By leveraging blockchain technology, we created a decentralized and tamper-proof system that increases user trust and system transparency. This chapter also discussed the selection of an appropriate DLT platform and the implementation of smart contracts to automate identity verification processes. We demonstrated how the combination of DLT and cryptographic techniques can provide a robust solution to the challenges of trust and privacy in identity management.

Chapter 5 - Implementation and Results We implemented the proposed system in two phases and validated its effectiveness through practical use cases. The first phase focused on a non-DLT enabled distributed identity provider, ensuring that the core functionalities of user authentication and identity verification were robust and user-friendly. In the second phase, we integrated DLT to enhance trust and transparency, utilizing blockchain technology to create an immutable record of identity transactions. The use cases demonstrated the system's applicability in real-world scenarios, showcasing its scalability and robustness. Specifically, we explored use cases in pandemic booking and smart city services, illustrating how our system can address current and future challenges in these domains.

6.1.3. Summary of Work Done

This thesis investigates the development and application of innovative solutions to address key challenges outlined in 1.3. The aim is to enhance existing methodologies and introduce novel frameworks within the realm of identity management. By integrating theoretical analysis with practical implementation, this work seeks to advance the field and establish a robust foundation for future research and development. More specifically, this thesis has focused on:

- Analyzed the limitations of existing centralized identity management systems and identified key challenges related to privacy, security, and trust.
- Proposed a conceptual framework for a privacy-preserving distributed identity management system, incorporating techniques such as zero-knowledge proofs (ZKP) and homomorphic encryption.
- Developed a DLT-enabled identity management system, providing a transparent and immutable record of identity transactions to enhance trust.
- Implemented the proposed system in two phases: a non-DLT enabled distributed identity provider and a DLT-enabled distributed identity provider.
- Validated the effectiveness of the proposed system through practical use cases, demonstrating its applicability in real-world scenarios such as pandemic booking and smart city services.

6.2. Future Work

This thesis was completed in November 2021. Although the time elapsed since its completion might suggest a missed window of opportunity, DLT technologies and privacy continue to be highly relevant. If a decision is made to promote and commercially integrate this work, several improvements and potential market projections could still be achieved. It is important to highlight that the projections from 2021 were more optimistic, driven by the initial enthusiasm and momentum around DLT and privacy technologies. In contrast, the current projections take a more conservative approach, reflecting a deeper and more realistic understanding of market conditions and the challenges associated with adoption. To forecast future adoption trends, this thesis employs the Bass diffusion model through Wolfram Alpha tool[1], considering key factors such as scalability, interoperability, and privacy enhancements.

Scalability and Performance Optimization Future research should focus on optimizing the scalability and performance of the DLT-enabled identity management system. As the number of users and transactions grows, it is crucial to ensure that the system can handle high volumes of data efficiently. This includes exploring more efficient consensus mechanisms, such as proof-of-stake or delegated proof-of-stake, which can offer better performance compared to traditional proof-of-work. Additionally, techniques like sharding and off-chain transactions can be investigated to enhance the system's throughput and reduce latency. As shown in Figure 6.1, the implementation of these techniques could significantly improve the transactions per second (TPS) by up to 150%.

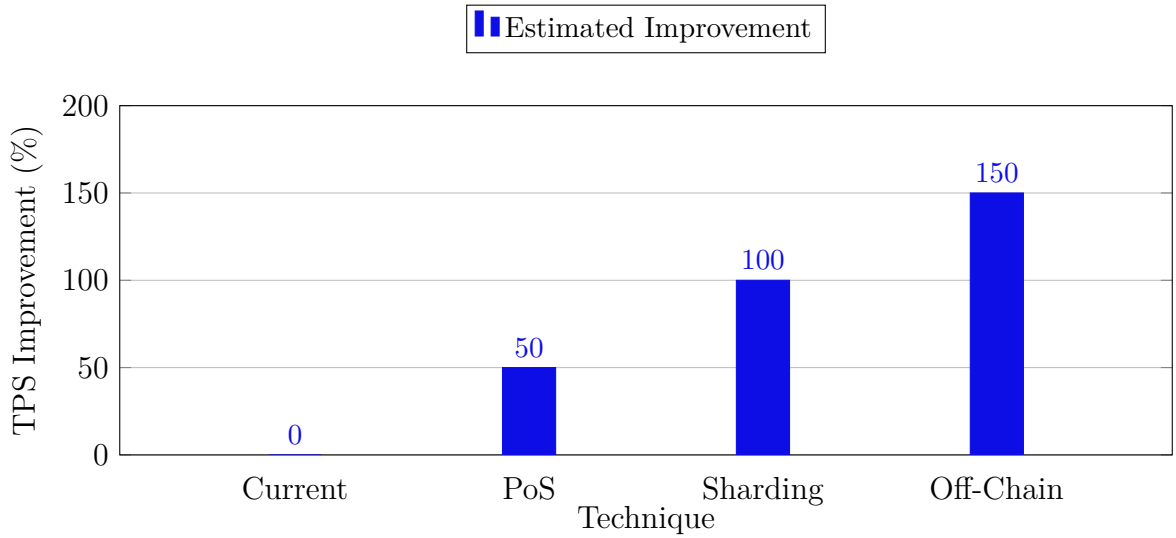


Figure 6.1: Estimated TPS Improvement with Different Techniques

[1] https://www.wolframalpha.com/

Interoperability with Existing Systems Ensuring seamless integration with existing identity management systems and standards is crucial for widespread adoption. Future work should address interoperability challenges and develop standardized protocols to facilitate integration. This includes aligning with standards like OAuth, OpenID Connect, and SAML, and developing APIs that allow easy integration with existing applications and services. Interoperability will enable the proposed system to complement and enhance current identity solutions rather than replacing them entirely. Figure 6.2 shows the projected increase in interoperability over time as these standards are adopted. The projections from 2021 were more optimistic due to the initial excitement around these technologies, whereas the current projections reflect a more cautious and measured approach.

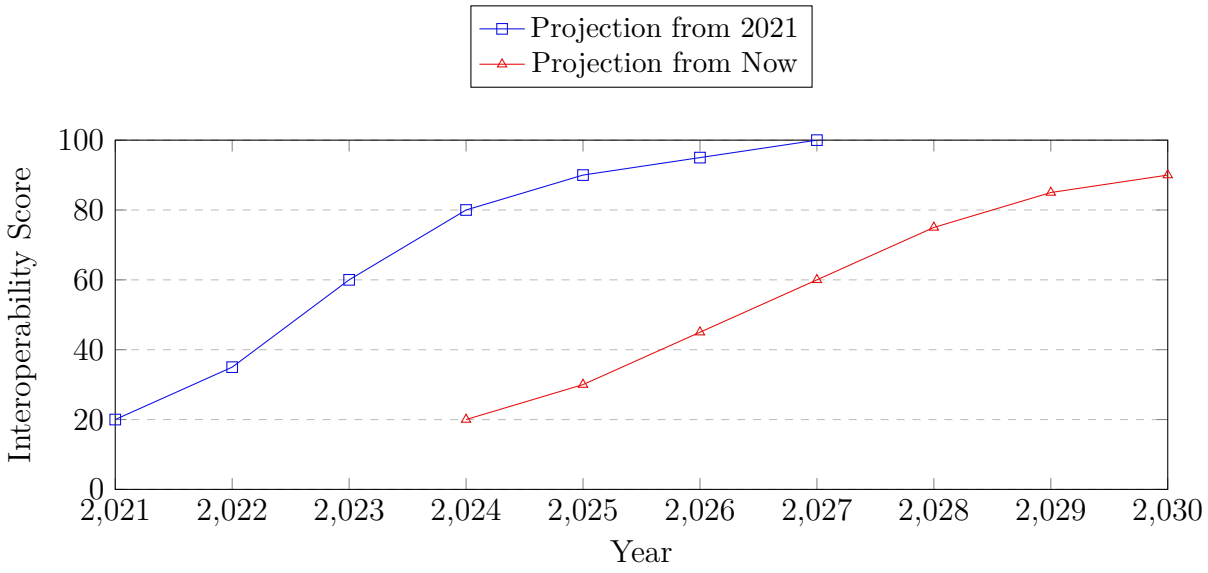


Figure 6.2: Projected Interoperability Score over Time

Enhanced Privacy Techniques Further development of advanced privacy-preserving techniques, such as secure multi-party computation (SMPC) and differential privacy, can provide additional layers of security and privacy for users. SMPC allows multiple parties to jointly compute a function over their inputs while keeping those inputs private. Differential privacy ensures that the removal or addition of a single data point does not significantly affect the outcome of any analysis, providing strong privacy guarantees. Integrating these techniques can enhance the robustness of the proposed system against various privacy threats. Figure 6.3 illustrates the projected improvement in privacy protection over time with these techniques. Again, the projections from 2021 were more optimistic compared to the more cautious current projections.

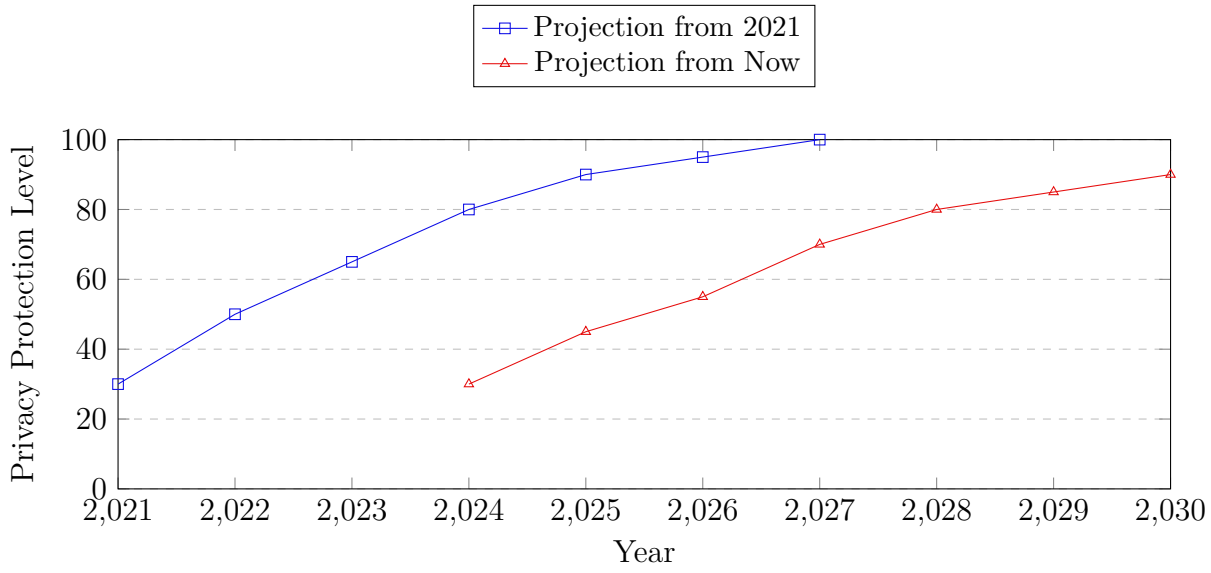


Figure 6.3: Projected Privacy Protection Level over Time

Regulatory Compliance and Legal Considerations As data protection regulations continue to evolve, future research should focus on ensuring that the proposed system remains compliant with global legal frameworks. This includes addressing issues related to data sovereignty, cross-border data transfers, and the right to be forgotten. Compliance with regulations such as the GDPR, CCPA, and emerging data protection laws in other regions is essential to build trust and ensure the system's legality. Future work should also explore the development of compliance tools that can automate the verification of regulatory requirements. Figure 6.4 shows the projected compliance readiness over the next few years. The projections from 2021 were more optimistic, reflecting a faster expected adoption, while the current projections are more conservative.

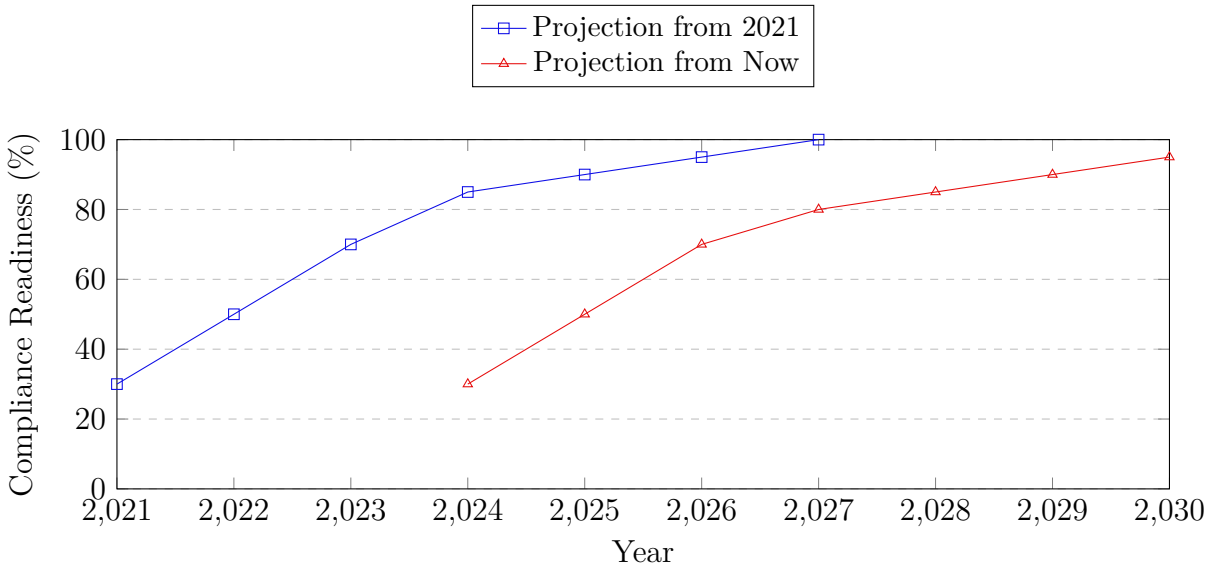


Figure 6.4: Projected Compliance Readiness over Time

Broader Use Case Validation Expanding the validation of the proposed system to include a wider range of use cases in different industries will help demonstrate its versatility and effectiveness. Potential sectors include finance, healthcare, and government services, where secure and privacy-preserving identity management is critical. For example, in finance, the system could be used for secure customer onboarding and KYC (Know Your Customer) processes. In healthcare, it could ensure the privacy of patient data while allowing secure access to medical records. In government services, the system could be used for secure digital voting and citizen identification.

User Experience and Adoption Investigating ways to enhance user experience and encourage adoption is essential. This includes developing intuitive user interfaces, providing comprehensive user education, and ensuring that the system meets the needs of diverse user groups. Future work should focus on designing user-friendly applications that simplify the interaction with the identity management system. Additionally, education and awareness campaigns can help users understand the benefits and functionalities of the system, fostering trust and encouraging widespread adoption.

Emerging Technologies Integration Future research should also explore the integration of emerging technologies, such as artificial intelligence (AI) and the Internet of Things (IoT), into the identity management system. AI can be used to enhance security through advanced threat detection and response mechanisms, while IoT devices can benefit from secure and reliable identity management. For instance, AI-driven analytics can identify and mitigate fraudulent activities in real-time, and IoT devices can securely authenticate and communicate within the network, ensuring the integrity and security of data exchanges.

A notable direction for future work involves analyzing and learning from the Cartera Digital Beta project in Spain[2], which aims to verify the ages of users accessing adult content online. It provides a significant case study for the integration of advanced privacy-preserving and verification technologies. This initiative aligns closely with the principles and technical underpinnings of the proposed DLT-enabled identity management system described in this thesis. The project leverages technologies such as OpenID for Verifiable Presentations and advanced cryptographic methods to ensure secure and private age verification.

The relevance of the Cartera Digital Beta project to our thesis lies in its application of privacy-preserving techniques and regulatory compliance frameworks, which are also central to our proposed solution. By analyzing this project, we can explore practical implementations of these technologies, understand the challenges faced, and identify opportunities for enhancing our identity management system. This project serves as a real-world example of how similar technologies can be deployed at scale, providing valuable insights for future research and development.

- **Privacy-Preserving Techniques:** The Cartera Digital Beta project emphasizes the use of privacy-preserving techniques to protect user data during the verification process. This aligns with the use of zero-knowledge proofs (ZKP) and homomorphic encryption in our proposed system to ensure that sensitive information is not exposed during authentication and verification.

- **Interoperability:** By adopting standards like OpenID for Verifiable Presentations, the Cartera Digital Beta project ensures interoperability with various systems and services. This mirrors the goal of our identity management system to seamlessly integrate with existing frameworks and standards, enhancing the usability and acceptance of the solution.

- **Regulatory Compliance:** The project is designed to comply with European regulations, such as GDPR, ensuring that the system adheres to strict data protection laws. This is similar to our focus on regulatory compliance, which is critical for building trust and ensuring the legal viability of the identity management system.

- **User Control and Consent:** The Cartera Digital Beta project empowers users with control over their personal data, allowing them to manage and consent to the sharing of their information. This approach is in line with the principles of self-sovereign identity (SSI) highlighted in our system, where users have autonomy over their identity attributes and credentials.

- **Scalability and Efficiency:** The project aims to efficiently handle a large number of verification requests, similar to the scalability objectives of our proposed system. By ensuring that the digital wallet can manage high volumes of data

[2] `https://digital.gob.es/especificaciones_tecnicas.html`

and user interactions, the project demonstrates the feasibility of deploying such solutions on a large scale.

- **Trust and Transparency:** Utilizing verifiable credentials and cryptographic methods, the Cartera Digital Beta project enhances trust and transparency in the verification process. This is akin to our use of Distributed Ledger Technology (DLT) to create an immutable and transparent record of identity transactions, fostering trust among all stakeholders.

While the Cartera Digital Beta project represents a significant step forward in the verification of age for accessing adult content online, it is not without its shortcomings. Several aspects of the project's implementation raise concerns, particularly regarding the use of whitelists for content management and the centralized nature of the Identity Provider (IdP).

Use of Whitelists for Content Management One of the primary weaknesses of the Cartera Digital Beta project is its reliance on whitelists to manage access to adult content. This approach involves maintaining a list of approved websites or services that are deemed appropriate for age-restricted access. While whitelists can be effective in controlling access, they also pose several risks:

- **Arbitrary Use and Censorship:** The use of whitelists can lead to arbitrary decisions about which sites are included or excluded, potentially resulting in censorship. This can be problematic if the criteria for inclusion on the whitelist are not transparent or if the process is subject to political or ideological influence. Such arbitrary control can undermine trust in the system and lead to accusations of unfairness or bias.

- **Maintenance and Updates:** Keeping a whitelist up-to-date requires continuous monitoring and maintenance. New adult content sites appear regularly, and ensuring that the whitelist reflects the current landscape of online content can be a daunting task. Any delays or failures in updating the whitelist can result in either over-blocking (denying access to legitimate sites) or under-blocking (allowing access to inappropriate sites).

- **False Sense of Security:** Relying on whitelists can create a false sense of security among users and administrators. There may be a tendency to believe that all whitelisted sites are safe and compliant, ignoring the possibility that some sites may change their content or practices over time. This complacency can lead to gaps in protection and potential exposure to inappropriate content.

Centralized Identity Provider (IdP) Another significant concern with the Cartera Digital Beta project is the centralized nature of the Identity Provider. Unlike our proposed distributed identity management system, which leverages a decentralized approach to enhance security and privacy, a centralized IdP has several drawbacks:

- **Single Point of Failure:** A centralized IdP represents a single point of failure. If the central IdP experiences technical issues, outages, or security breaches, the entire age verification system could be compromised. This vulnerability can lead to downtime and a loss of trust among users and service providers.
- **Privacy Risks:** Centralized IdPs can collect and store large amounts of personal data, making them attractive targets for hackers. In the event of a data breach, sensitive information about users, including their age verification status and possibly their browsing habits, could be exposed. This centralization of data poses significant privacy risks that are mitigated in a decentralized system.
- **Lack of User Control:** In a centralized system, users have limited control over their personal data. They must rely on the IdP to manage and protect their information, which can be problematic if the IdP's policies or practices do not align with the user's privacy preferences. In contrast, a decentralized system empowers users by giving them more control over their identity attributes and how they are shared.
- **Scalability Issues:** As the number of users grows, a centralized IdP may struggle to scale effectively. Handling large volumes of verification requests and managing the associated data can become increasingly challenging. Decentralized systems, on the other hand, can scale more efficiently by distributing the load across multiple nodes.

While the Cartera Digital Beta project makes significant strides in age verification for accessing adult content, it falls short in several critical areas. The reliance on whitelists for content management introduces risks of arbitrary use and censorship, maintenance challenges, and a false sense of security. Additionally, the centralized nature of the IdP poses privacy risks, creates a single point of failure, limits user control, and may face scalability issues as user numbers increase.

Future work on our proposed DLT-enabled identity management system can draw valuable lessons from these shortcomings. By focusing on a decentralized approach, we can enhance privacy, security, and user control, while also addressing scalability concerns. Integrating more robust and transparent mechanisms for content management, rather than relying on whitelists, can further improve the system's fairness and reliability. Through these improvements, we can develop a more resilient and trustworthy solution for digital identity management and age verification.

In conclusion, this thesis has made significant contributions to the field of digital identity management by proposing and validating a novel system that enhances privacy, security, and trust. Future work will build on this foundation to further refine and expand the capabilities of the system, ensuring its relevance and effectiveness in an ever-evolving digital landscape.

CHAPTER 7

Bibliography

Bibliography

[1] N. Notario, A. Crespo, A. Skarmeta, J. Bernal, and J. L. Cánovas, "Aries: Reliable european identity ecosystem," *ERCIM News*, no. 109, 2020.

[2] P. Bichsel, J. Camenisch, M. Dubovitskaya, R. R. Enderlein, S. Krenn, I. Krontiris, A. Lehmann, G. Neven, C. Paquin, F. Preiss, K. Rannenberg, and A. Sabouri, "An architecture for privacy-abcs," in *Attribute-based Credentials for Trust: Identity in the Information Society*, pp. 11–78, Springer, 2015.

[3] P. Voigt and A. Von dem Bussche, "The eu general data protection regulation (gdpr)," *A Practical Guide, 1st Ed., Cham: Springer International Publishing*, vol. 10, p. 3152676, 2017.

[4] J. Isaak and M. J. Hanna, "User data privacy: Facebook, cambridge analytica, and privacy protection," *Computer*, vol. 51, no. 8, pp. 56–59, 2018.

[5] R. Maull, P. Godsiff, C. Mulligan, A. Brown, and B. Kewell, "Distributed ledger technology: Applications and implications," *Strategic Change*, vol. 26, no. 5, pp. 481–489, 2017.

[6] D. Tapscott and A. Tapscott, *Blockchain revolution: how the technology behind bitcoin is changing money, business, and the world*. Penguin, 2016.

[7] S. Nakamoto, "Bitcoin: A peer-to-peer electronic cash system," tech. rep., Manubot, 2019.

[8] J. B. Bernabe, J. L. Canovas, J. L. Hernandez-Ramos, R. T. Moreno, and A. Skarmeta, "Privacy-preserving solutions for blockchain: review and challenges," *IEEE Access*, vol. 7, pp. 164908–164940, 2019.

[9] A. Tobin and D. Reed, "The inevitable rise of self-sovereign identity," *The Sovrin Foundation*, vol. 29, no. 2016, 2016.

[10] "European self sovereign identity framework," Jun 2019. https://www.eesc.europa.eu/en/news-media/presentations/european-self-sovereign-identity-framework.

[11] K. Rannenberg, J. Camenisch, and A. Sabouri, "Attribute-based credentials for trust," *Identity in the Information Society, Springer*, 2015.

[12] J. Camenisch and A. Lysyanskaya, "An efficient system for non-transferable anonymous credentials with optional anonymity revocation," in *International conference on the theory and applications of cryptographic techniques*, pp. 93–118, Springer, 2001.

[13] S. ShoCard, "Travel identity of the future," 2016. https://whitepaper.uport.me/uPort_ whitepaper_DRAFT20170221.pdf.

[14] "uport project," tech. rep. https://github.com/uport-project/specs.

[15] R. T. Moreno, J. B. Bernabe, A. Skarmeta, M. Stausholm, T. K. Frederiksen, N. Martínez, N. Ponte, E. Sakkopoulos, and A. Lehmann, "Olympus: Towards oblivious identity management for private and user-friendly services," in *2019 Global IoT Summit (GIoTS)*, pp. 1–6, IEEE, 2019.

[16] R. Torres Moreno, J. Bernal Bernabe, J. Garcia Rodriguez, T. Kasper Frederiksen, M. Stausholm, N. Martínez, E. Sakkopoulos, N. Ponte, and A. Skarmeta, "The olympus architecture—oblivious identity management for private user-friendly services," *Sensors*, vol. 20, no. 3, p. 945, 2020.

[17] R. T. Moreno, J. G. Rodríguez, C. T. López, J. B. Bernabe, and A. Skarmeta, "Olympus: A distributed privacy-preserving identity management system," in *2020 Global Internet of Things Summit (GIoTS)*, pp. 1–6, IEEE, 2020.

[18] J. García-Rodríguez, R. Torres Moreno, J. Bernal Bernabé, and A. Skarmeta, "Towards a standardized model for privacy-preserving verifiable credentials," in *The 16th International Conference on Availability, Reliability and Security*, pp. 1–6, 2021.

[19] J. García-Rodríguez, R. T. Moreno, J. B. Bernabe, and A. Skarmeta, "Implementation and evaluation of a privacy-preserving distributed abc scheme based on multi-signatures," *Journal of Information Security and Applications*, vol. 62, p. 102971, 2021.

[20] R. T. Moreno, J. García-Rodríguez, J. B. Bernabé, and A. Skarmeta, "A trusted approach for decentralised and privacy-preserving identity management," *IEEE Access*, vol. 9, pp. 105788–105804, 2021.

[21] J. B. Bernabe, M. David, R. T. Moreno, J. P. Cordero, S. Bahloul, and A. Skarmeta, "Aries: Evaluation of a reliable and privacy-preserving european identity management framework," *Future Generation Computer Systems*, vol. 102, pp. 409–425, 2020.

[22] S. Daoudagh, E. Marchetti, V. Savarino, J. B. Bernabe, J. García-Rodríguez, R. T. Moreno, J. A. Martinez, and A. F. Skarmeta, "Data protection by design in the context of smart cities: A consent and access control proposal," *Sensors*, vol. 21, no. 21, p. 7154, 2021.

[23] J. B. Bernabe, J. García-Rodríguez, S. Krenn, V. Liagkou, A. Skarmeta, and R. Torres, "Privacy-preserving identity management and applications to academic degree verification," in *IFIP International Summer School on Privacy and Identity Management*, pp. 33–46, Springer, 2022.

[24] J. B. Bernabe, R. Torres, D. Martin, A. Crespo, A. Skarmeta, D. Fortune, J. Lodge, T. Oliveira, M. Silva, S. Martin, *et al.*, "An overview on aries: Reliable european identity ecosystem."

[25] T. K. Frederiksen, J. Hesse, A. Lehmann, and R. Torres Moreno, "Identity management: State of the art, challenges and perspectives," in *IFIP International Summer School on Privacy and Identity Management*, pp. 45–62, Springer, 2019.

[26] D. Recordon and D. Reed, "Openid 2.0: a platform for user-centric identity management," in *Proceedings of the second ACM workshop on Digital identity management*, pp. 11–16, 2006.

[27] E. Hammer-Lahav, D. Recordon, and D. Hardt, "The oauth 1.0 protocol," tech. rep., RFC 5849, April, 2010.

[28] D. Hardt *et al.*, "The oauth 2.0 authorization framework," 2012.

[29] J. Hughes and E. Maler, "Security assertion markup language (saml) v2. 0 technical overview," *OASIS SSTC Working Draft sstc-saml-tech-overview-2.0-draft-08*, vol. 13, 2005.

[30] R. Housley, W. Ford, W. Polk, D. Solo, *et al.*, "Internet x. 509 public key infrastructure certificate and crl profile," tech. rep., RFC 2459, January, 1999.

[31] P. Consortium *et al.*, "Privacy and identity management in europe for life (primelife)," tech. rep., FP7-ICT-2007-1, Version 3, 09/10/2007, Grant Agreement GA.

[32] "Flickr's authorization api." `https://www.flickr.com/services/api/auth.oauth.html`.

[33] "Authsub in the google data protocol client libraries." `https://developers.google.com/gdata/docs/auth/authsub`.

[34] N. Sakimura, J. Bradley, M. Jones, B. De Medeiros, and C. Mortimore, "Openid connect core 1.0," *The OpenID Foundation*, p. S3, 2014.

[35] D. Cooper, S. Santesson, S. Farrell, S. Boeyen, R. Housley, W. T. Polk, *et al.*, "Internet x. 509 public key infrastructure certificate and certificate revocation list (crl) profile.," *RFC*, vol. 5280, pp. 1–151, 2008.

[36] P. Bichsel, J. Camenisch, M. Dubovitskaya, R. R. Enderlein, S. Krenn, I. Krontiris, A. Lehmann, G. Neven, C. Paquin, F.-S. Preiss, *et al.*, "An architecture for privacy-abcs," in *Attribute-Based Credentials for Trust*, pp. 11–78, Springer, 2015.

[37] J. Camenisch and A. Lysyanskaya, "An efficient system for non-transferable anonymous credentials with optional anonymity revocation," in *International conference on the theory and applications of cryptographic techniques*, pp. 93–118, Springer, 2001.

[38] D. Chaum, "Blind signatures for untraceable payments," in *Advances in cryptology*, pp. 199–203, Springer, 1983.

[39] J. L. Camenisch, J.-M. Piveteau, and M. A. Stadler, "Blind signatures based on the discrete logarithm problem," in *Workshop on the Theory and Application of of Cryptographic Techniques*, pp. 428–432, Springer, 1994.

[40] U. Feige, A. Fiat, and A. Shamir, "Zero-knowledge proofs of identity," *Journal of cryptology*, vol. 1, no. 2, pp. 77–94, 1988.

[41] J. Camenisch, S. Mödersheim, and D. Sommer, "A formal model of identity mixer," in *International Workshop on Formal Methods for Industrial Critical Systems*, pp. 198–214, Springer, 2010.

[42] C. Paquin and G. Zaverucha, "U-prove cryptographic specification v1. 1," *Technical Report, Microsoft Corporation*, 2011.

[43] P. Persiano and I. Visconti, "An anonymous credential system and a privacy-aware pki," in *Australasian Conference on Information Security and Privacy*, pp. 27–38, Springer, 2003.

[44] L. M. Bach, B. Mihaljevic, and M. Zagar, "Comparative analysis of blockchain consensus algorithms," in *2018 41st International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO)*, pp. 1545–1550, Ieee, 2018.

[45] S. M. H. Bamakan, A. Motavali, and A. B. Bondarti, "A survey of blockchain consensus algorithms performance evaluation criteria," *Expert Systems with Applications*, vol. 154, p. 113385, 2020.

[46] "Nem whitepaper," Feb 2021. `https://www.allcryptowhitepapers.com/nem-whitepaper/`.

[47] L. Baird, "The swirlds hashgraph consensus algorithm: Fair, fast, byzantine fault tolerance," *Swirlds Tech Reports SWIRLDS-TR-2016-01, Tech. Rep*, vol. 34, 2016.

[48] M. Castro, B. Liskov, *et al.*, "Practical byzantine fault tolerance," in *OsDI*, vol. 99, pp. 173–186, 1999.

[49] F. M. Benčić and I. P. Žarko, "Distributed ledger technology: Blockchain compared to directed acyclic graph," in *2018 IEEE 38th International Conference on Distributed Computing Systems (ICDCS)*, pp. 1569–1570, IEEE, 2018.

[50] S. Popov, "The tangle," *White paper*, vol. 1, no. 3, 2018.

[51] W. F. Silvano and R. Marcelino, "Iota tangle: A cryptocurrency to communicate internet-of-things data," *Future generation computer systems*, vol. 112, pp. 307–319, 2020.

[52] E. Harris-Braun, N. Luck, and A. Brock, "Holochain-scalable agentcentric distributed computing," *Alpha*, vol. 1, pp. 1–14, 2018.

[53] "Radix dlt: Radically different defi."

[54] J. Hellings, D. P. Hughes, J. Primero, and M. Sadoghi, "Cerberus: Minimalistic multi-shard byzantine-resilient transaction processing," *arXiv preprint arXiv:2008.04450*, 2020.

[55] F. Cäsar, D. P. Hughes, J. Primero, and S. J. Thornton, "A parallelized bft consensus protocol for radix," 2020.

[56] L. Alber, S. More, S. Mödersheim, and A. Schlichtkrull, "Adapting the tpl trust policy language for a self-sovereign identity world," *Open Identity Summit 2021*, 2021.

[57] A. Kosba, A. Miller, E. Shi, Z. Wen, and C. Papamanthou, "Hawk: The blockchain model of cryptography and privacy-preserving smart contracts," in *2016 IEEE symposium on security and privacy (SP)*, pp. 839–858, IEEE, 2016.

[58] D. Hopwood, S. Bowe, T. Hornby, and N. Wilcox, "Zcash protocol specification," *GitHub: San Francisco, CA, USA*, 2016.

[59] I. Miers, C. Garman, M. Green, and A. D. Rubin, "Zerocoin: Anonymous distributed e-cash from bitcoin," in *2013 IEEE Symposium on Security and Privacy*, pp. 397–411, IEEE, 2013.

[60] D. Khovratovich and J. Law, "Sovrin: digital identities in the blockchain era," *Github Commit by jasonalaw October*, vol. 17, 2017.

[61] The Hyperledger Project, "Hyperledger aries." `https://www.hyperledger.org/projects/aries`, 2024.

[62] V. Dhillon, D. Metcalf, and M. Hooper, "The hyperledger project," in *Blockchain enabled applications*, pp. 139–149, Springer, 2017.

[63] D. Reed, M. Sporny, D. Longley, C. Allen, R. Grant, M. Sabadello, and J. Holt, "Decentralized identifiers (dids) v1. 0," *Draft Community Group Report*, 2020.

[64] J. Camenisch and E. Van Herreweghen, "Design and implementation of the idemix anonymous credential system," in *Proceedings of the 9th ACM Conference on Computer and Communications Security*, pp. 21–30, 2002.

[65] C. Berger and H. P. Reiser, "Scaling byzantine consensus: A broad analysis," in *Proceedings of the 2nd workshop on scalable and resilient infrastructures for distributed ledgers*, pp. 13–18, 2018.

[66] C. Dannen, *Introducing Ethereum and solidity*, vol. 1. Springer, 2017.

[67] G. Wood *et al.*, "Ethereum: A secure decentralised generalised transaction ledger," *Ethereum project yellow paper*, vol. 151, no. 2014, pp. 1–32, 2014.

[68] S. Muralidharan and H. Ko, "An interplanetary file system (ipfs) based iot framework," in *2019 IEEE international conference on consumer electronics (ICCE)*, pp. 1–2, IEEE, 2019.

[69] World Wide Web Consortium (W3C), "Verifiable credentials data model 1.0," 2019. Accessed: 2024-05-04.

[70] X. Yi, R. Paulet, E. Bertino, X. Yi, R. Paulet, and E. Bertino, *Homomorphic encryption*. Springer, 2014.

[71] J. Camenisch, S. Krenn, and V. Shoup, "A framework for practical universally composable zero-knowledge protocols," in *International Conference on the Theory and Application of Cryptology and Information Security*, pp. 449–467, Springer, 2011.

[72] C. Zhao, S. Zhao, M. Zhao, Z. Chen, C.-Z. Gao, H. Li, and Y.-a. Tan, "Secure multi-party computation: theory, practice and applications," *Information Sciences*, vol. 476, pp. 357–372, 2019.

[73] K. O'Flaherty, "Collection 1 breach – how to find out if your password has been stolen," *Forbes*, 2019.

[74] L. H. Newman, "Equifax officially has no excuse," *Wired*.

[75] N. Hong, L. Hoffman, and A. Andriotis, "Capital one reports data breach affecting 100 million customers, applicants," *The Wall Street Journal*.

[76] Y. G. Desmedt, "Threshold cryptography," *European Transactions on Telecommunications*, vol. 5, no. 4, pp. 449–458, 1994.

[77] J. Camenisch and A. Lysyanskaya, "Signature schemes and anonymous credentials from bilinear maps," in *Annual international cryptology conference*, pp. 56–72, Springer, 2004.

[78] O. Goldreich, "Secure multi-party computation," *Manuscript. Preliminary version*, vol. 78, p. 110, 1998.

[79] M. J. Freedman, Y. Ishai, B. Pinkas, and O. Reingold, "Keyword search and oblivious pseudorandom functions," in *Theory of Cryptography Conference*, pp. 303–324, Springer, 2005.

[80] R. Gennaro, S. Goldfeder, and B. Ithurburn, "Fully distributed group signatures (2019)," *URL https://www. orbs. com/wp-content/uploads/2019/04/Crypto_Group_ signatures-2. pdf*.

[81] T. K. Frederiksen, Y. Lindell, V. Osheter, and B. Pinkas, "Fast distributed rsa key generation for semi-honest and malicious adversaries," in *Annual International Cryptology Conference*, pp. 331–361, Springer, 2018.

[82] Y. Frankel, P. D. MacKenzie, and M. Yung, "Robust efficient distributed rsa-key generation," in *Proceedings of the thirtieth annual ACM symposium on Theory of computing*, pp. 663–672, 1998.

[83] D. L. Chaum, "Untraceable electronic mail, return addresses, and digital pseudonyms," *Communications of the ACM*, vol. 24, no. 2, pp. 84–90, 1981.

[84] D. Chaum, "Security without identification: Transaction systems to make big brother obsolete," *Communications of the ACM*, vol. 28, no. 10, pp. 1030–1044, 1985.

[85] D. Boneh, X. Boyen, and H. Shacham, "Short group signatures," in *Annual international cryptology conference*, pp. 41–55, Springer, 2004.

[86] D. Pointcheval and O. Sanders, "Short randomizable signatures," in *Cryptographers' Track at the RSA Conference*, pp. 111–126, Springer, 2016.

[87] A. Sonnino, M. Al-Bassam, S. Bano, S. Meiklejohn, and G. Danezis, "Coconut: Threshold issuance selective disclosure credentials with applications to distributed ledgers," *arXiv preprint arXiv:1802.07344*, 2018.

[88] S. Micali, K. Ohta, and L. Reyzin, "Accountable-subgroup multisignatures," in *Proceedings of the 8th ACM Conference on Computer and Communications Security*, pp. 245–254, 2001.

[89] J. Camenisch, M. Drijvers, A. Lehmann, G. Neven, and P. Towa, "Short threshold dynamic group signatures," in *International Conference on Security and Cryptography for Networks*, pp. 401–423, Springer, 2020.

[90] "European h2020 project cybersec4europe," Jan 2022. `https://cybersec4europe.eu/`.

[91] C. Baum, T. Frederiksen, J. Hesse, A. Lehmann, and A. Yanai, "Pesto: proactively secure distributed single sign-on, or how to trust a hacked server," in *2020 IEEE European Symposium on Security and Privacy (EuroS&P)*, pp. 587–606, IEEE, 2020.

[92] M. Jones, J. Bradley, and N. Sakimura, "Json web token (jwt)," tech. rep., 2015.

[93] D. W. Chadwick, R. Laborde, A. Oglaza, R. Venant, S. Wazan, and M. Nijjar, "Improved identity management with verifiable credentials and fido," *IEEE Communications Standards Magazine*, vol. 3, no. 4, pp. 14–20, 2019.

[94] M. Scott, "The apache milagro crypto library."

[95] "Hyperledger indy," tech. rep. https://indy.readthedocs.io/en/latest/.

[96] N. Szabo, "Smart contracts," *Unpublished manuscript*, 1994.

[97] K. Christidis and M. Devetsikiotis, "Blockchains and smart contracts for the internet of things," *Ieee Access*, vol. 4, pp. 2292–2303, 2016.

[98] C. Cachin *et al.*, "Architecture of the hyperledger blockchain fabric," in *Workshop on distributed cryptocurrencies and consensus ledgers*, vol. 310, Chicago, IL, 2016.

[99] P. Thakkar, S. Nathan, and B. Viswanathan, "Performance benchmarking and optimizing hyperledger fabric blockchain platform," in *2018 IEEE 26th International Symposium on Modeling, Analysis, and Simulation of Computer and Telecommunication Systems (MASCOTS)*, pp. 264–276, IEEE, 2018.

[100] H. Sukhwani, N. Wang, K. S. Trivedi, and A. Rindos, "Performance modeling of hyperledger fabric (permissioned blockchain network)," in *2018 IEEE 17th International Symposium on Network Computing and Applications (NCA)*, pp. 1–8, IEEE, 2018.

[101] M. Kuzlu, M. Pipattanasomporn, L. Gurses, and S. Rahman, "Performance analysis of a hyperledger fabric blockchain framework: throughput, latency and scalability," in *2019 IEEE international conference on blockchain (Blockchain)*, pp. 536–540, IEEE, 2019.

[102] Node.js Foundation, "Node.js documentation," 2023. Accessed: 2023-05-03.

[103] S. Thorgersen and P. I. Silva, *Keycloak-identity and access management for modern applications: harness the power of Keycloak, OpenID Connect, and OAuth 2.0 protocols to secure applications.* Packt Publishing Ltd, 2021.

[104] F. Cirillo, G. Solmaz, E. L. Berz, M. Bauer, B. Cheng, and E. Kovacs, "A standard-based open source iot platform: Fiware," *IEEE Internet of Things Magazine*, vol. 2, no. 3, pp. 12–18, 2019.

[105] X. Wang, H. Zhao, and J. Zhu, "Grpc: A communication cooperation mechanism in distributed systems," *ACM SIGOPS Operating Systems Review*, vol. 27, no. 3, pp. 75–86, 1993.