Universidad de Buenos Aires

Facultades de Ciencias Económicas,
Cs. Exactas y Naturales e Ingeniería

Carrera de Especialización en Seguridad Informática

Trabajo Final de la Especialización

Título: Gestión de Identidad Digital Descentralizada (SSI) con Blockchain

Autor: Ing. Marcos H. Martínez

Tutor del Trabajo Final: Dr. Hugo Scolnik

Año de presentación 2020

Cohorte 2019

Declaración Jurada de origen de los contenidos

Por medio de la presente, el autor manifiesta conocer y aceptar el Reglamento de Trabajos Finales vigente y se hace responsable que la totalidad de los contenidos del presente documento son originales y de su creación exclusiva, o bien pertenecen a terceros u otras fuentes, que han sido adecuadamente referenciados y cuya inclusión no infringe la legislación Nacional e Internacional de Propiedad Intelectual.

Martínez Marcos Hernán
DNI 31.070.807
FIRMADO

Resumen

El presente trabajo busca introducir al lector en el movimiento conocido como Identidad Digital Autosoberana, Descentralizada o SSI por sus siglas en inglés (Self Sovereign Identity).

Este modelo es realmente joven, surgió a partir del año 2016, su difusión se realiza más que nada a través de la comunidad de SSIMeetup y al momento de escribir este trabajo, integrantes importantes de la misma, recién estiman para el verano del 2020 finalizar el primer libro sobre este movimiento. Por lo tanto, este documento intenta promover: su filosofía, arquitectura y el desarrollo de algunos casos de uso que, a pesar de no tener productos tecnológicos terminados, seguramente y con el tiempo, se producirá un cambio disruptivo en la forma de tratar nuestra identidad.

Con el fin de acatar los conceptos relevantes del modelo, se realiza una PoC a través de un escenario. Si bien existen varias versiones del mismo, se escogió la recomendada para lectores que recién se inician en SSI.

Por último, se explican los principales desafíos que presenta este ecosistema para que sea utilizado a gran escala. Los cuales son naturales al ser un movimiento emergente; por lo tanto, habrá que esperar a que el tiempo normalice paulatinamente esta situación.

Palabras Clave: SSI, Identidad Digital Autosoberana, Identidad Digital Descentralizada, Blockchain, Hyperledger Indy, Credenciales Verificables.

Índice de contenidos

DECLARACIÓN JURADA DE ORIGEN DE LOS CONTENIDOS I

RESUMEN II

ÍNDICE DE CONTENIDOS III

ÍNDICE DE ILUSTRACIONES V

PRÓLOGO VII

NÓMINA DE ABREVIATURAS VIII

INTRODUCCIÓN 1

1 CAPÍTULO 1: INTRODUCCIÓN A IDENTIDAD DIGITAL DESCENTRALIZADA (SSI) 3

1.1 Identidad digital 3

1.2 Modelos de identidad 5

1.2.1 Modelo de Identidad Centralizada 5

1.2.2 El modelo de Identidad Federado 6

1.2.3 El modelo de Identidad Descentralizado 7

1.3 ¿Qué es SSI? 9

1.4 Bloques principales de SSI 12

1.4.1 Credenciales verificables 13

1.4.2 Emisores, titulares y verificadores 16

1.4.3 Billeteras Digitales (Wallet) 17

1.4.4 Agentes digitales y centros 18

1.4.5 Identificadores descentralizados (DID) 19

1.4.6 Blockchains 29

1.4.7 Marcos de gobernanza 30

1.4.8 Sistema Gestión de Claves Descentralizado (DKMS) 32

1.5 Arquitectura SSI 36

1.5.1 Capa 1: Claves e identificadores 36
1.5.2 Capa 2: Interfaces y comunicaciones seguras 37
1.5.3 Capa 3: Credenciales 38
1.5.4 Capa 4: Gobierno 39

2 CAPÍTULO 2: CASOS DE USO SSI 40

2.1 Casos de Uso en Argentina 40

2.2 Demostración de un escenario de SSI 43
2.2.1 Estableciendo Relaciones entre Alice y Faber 46
2.2.2 Emitiendo credencial de Faber a Alice 49
2.2.3 Solicitar empleo a Acme 53

3 CAPÍTULO 3: DESAFÍOS DE SSI 59

3.1 Construyendo el nuevo ecosistema SSI 59

3.2 Gestión descentralizada de claves (DKMS) 59

3.3 Acceso sin conexión 60

3.4 La desconexión del mundo Blockchain del mundo SSI 60

3.5 La innovación tecnológica como marketing. 61

3.6 La falta de una red de apoyo para financiar el emprendimiento en el mundo SSI 61

CONCLUSIÓN 62

BIBLIOGRAFÍA 64

Índice de ilustraciones

ILUSTRACIÓN 1 ESQUEMA DE IDENTIDAD DIGITAL 4
ILUSTRACIÓN 2 MODELO CENTRALIZADO 5
ILUSTRACIÓN 3 MODELO DE IDENTIDAD FEDERADO 6
ILUSTRACIÓN 4 MODELO DE IDENTIDAD DESCENTRALIZADO 7
ILUSTRACIÓN 5 : IDENTIDAD DIGITAL DESCENTRALIZADA 12
ILUSTRACIÓN 6 MAPEO DE CREDENCIAL FÍSICA A 3 COMPONENTES DE W3C 14
ILUSTRACIÓN 7 ROLES EN LAS CREDENCIALES VERIFICABLES 16
ILUSTRACIÓN 8 COMUNICACIÓN ENTRE AGENTES Y BILLETERAS DE BORDE 18
ILUSTRACIÓN 9 DIAGRAMA DE AGENTES Y BILLETERAS EN SSI 18
ILUSTRACIÓN 11 EJEMPLO DE UN DID DE SOVRIN 20
ILUSTRACIÓN 12 EJEMPLO FRAGMENTO DE DID 21
ILUSTRACIÓN 13 EJEMPLO DID UTILIZANDO QUERY 21
ILUSTRACIÓN 14 RELACIÓN ENTRE DID, SU DOCUMENTO Y SUJETO 22
ILUSTRACIÓN 15 EJEMPLO DE DOCUMENTO DID 23
ILUSTRACIÓN 16 EJEMPLO DE 5 MÉTODOS DID 24
ILUSTRACIÓN 17 VERIFICACIÓN DE UNA FIRMA EN UNA CREDENCIAL 25
ILUSTRACIÓN 18 DID AUTH A ALTO NIVEL 27
ILUSTRACIÓN 19 ARQUITECTURA 1 DID AUTH: PÁGINA WEB Y APLICACIÓN MÓVIL. 28
ILUSTRACIÓN 20 SEGUNDO CÍRCULO DE CONFIANZA 30
ILUSTRACIÓN 21 ARQUITECTURA DKMS 33
ILUSTRACIÓN 22 LA PILA DE SSI DE CUATRO CAPAS 36
ILUSTRACIÓN 23 PROYECTOS DE PAÍSES EN IBEROAMÉRICA 41
ILUSTRACIÓN 24 IMAGEN ENTORNO DOCKER 44
ILUSTRACIÓN 25 ALGUNOS PUERTOS MOSTRADOS EN EL DOCKER 45
ILUSTRACIÓN 26 INTERFACE DEL AGENTE ALICE 46
ILUSTRACIÓN 27 EJEMPLO DE CONEXIÓN A TRAVÉS DE QR 47
ILUSTRACIÓN 28 CREAR UNA RELACIÓN CON FABER A TRAVÉS DE SU DID 48
ILUSTRACIÓN 29 SOLICITUD DE NOMBRE DEL AGENTE FABER A ALICE 48
ILUSTRACIÓN 30 SOLICITUD DE NOMBRE DEL AGENTE ALICE A FABER 49
ILUSTRACIÓN 31 SECCIÓN RELACIONES DE FABER CON EL NOMBRE DE ALICE ACTUALIZADO 49
ILUSTRACIÓN 32 SECCIÓN RELACIONES DE ALICE CON EL NOMBRE DE FABER ACTUALIZADO 49
ILUSTRACIÓN 33 EJEMPLO ESQUEMA DE TRANSCRIPCIÓN 50
ILUSTRACIÓN 34 EJEMPLO DE DEFINICIÓN DE CREDENCIAL 51
ILUSTRACIÓN 35 EMISIÓN DE CREDENCIAL HACIA ALICE 52
ILUSTRACIÓN 36 MENSAJE DE OFRECIMIENTO DE CREDENCIAL DE FABER A ALICE 52
ILUSTRACIÓN 37 CREDENCIAL ACADÉMICA DE ALICE 53
ILUSTRACIÓN 38 SOLICITUD DE CONEXIÓN DE ALICE A ACME 54

ILUSTRACIÓN 39 SOLICITUD PRUEBA DE NOMBRE DE ACME PARA ALICE 54
ILUSTRACIÓN 40 SOLICITUD PRUEBA DE NOMBRE DE ALICE PARA ACME 54
ILUSTRACIÓN 41 COPIANDO SOLICITUD DE PRUEBA DE CREDENCIALES FABER-ACME 55
ILUSTRACIÓN 42 SOLICITUD DE PRUEBA DE ACME A ALICE A TRAVÉS DE MÉTODO MANUAL 56
ILUSTRACIÓN 43 MENSAJE DE SOLICITUD DE PRUEBA DE ACME A ALICE 57
ILUSTRACIÓN 44 PRUEBAS DE ALICE EN ACME 57
ILUSTRACIÓN 45 ACME VALIDA PRUEBA PROPORCIONADA POR ALICE 58

Prólogo

En principio, quisiera agradecer a Dios que siempre está presente, a mis padres, Aída Noemí Morán y Ricardo Martínez debido a que son un apoyo incondicional en cada paso que doy. Siempre me brindaron sus fuerzas y tiempo para emprender lo que quisiese.

A mi tutor, el doctor Hugo Scolnik por sus correcciones y aportes en este trabajo.

A las personas que se cruzaron en mi vida, brindando aliento y también su punto de vista en temas que quizás no eran de su interés pero que aportaron mucho.

Por último, a todos aquellos que escuchando o estando simplemente hicieron que este trabajo sea posible.

A todos, gracias.

Nómina de abreviaturas

SAML: Security Assertion Markup Language. lenguaje de marcado para confirmaciones de seguridad.

PoC**:** Proof of Concept, prueba de concepto.

DLT: Distributed Ledger Technology, tecnologías de registro contable distribuido.

TLS: Transport Layer Security, seguridad en la capa de transporte.

W3C: World Wide Web Consortium, consorcio de la www.

URN: Uniform Resource Name, nombre de recurso uniforme.

URL: Uniform Resource Locator, localizador de recursos uniforme.

RSA: Rivest, Shamir y Adleman.

API: Application Programming Interface, interfaz de programación de aplicaciones.

HTTP: Hypertext Transfer Protocol, protocolo de transferencia de hipertexto.

SMTP: Simple Mail Transfer Protocol, protocolo para transferencia simple de correo.

HTTPS: Hypertext Transfer Protocol Secure, protocolo seguro de transferencia de hipertexto.

QR: Quick Response, respuesta rápida.

JSON: JavaScript Object Notation, notación de objeto de JavaScript.

GDPR: General Data Protection Regulation, reglamento general de protección de datos.

PKI: Public Key Infrastructure, infraestructura de clave pública.

JSON-LD: JavaScript Object Notation for Linked Data, notación de objeto de JavaScript para datos enlazados.

JWT: JSON Web Token, notación de objeto de JavaScript para token web.

IDP: Identity Provider, proveedor de identidad.

Oauth: Open Authorization

Introducción

A diario, utilizamos servicios online, estos manejan identidades digitales, las cuales son representaciones de nosotros mismos. Cada vez que queremos acceder a ellos debemos emplearla, ya sea a través del propio sistema o de un IDP.

Este modelo, denominado centralizado, presenta muchos problemas debido a que no somos dueños de nuestra identidad, delegando la responsabilidad de custodia y gestión de nuestros datos personales a un tercero, no necesariamente confiable. Lo cual, puede generar pérdida, falsificación o tratamiento de la información para otros usos de los cuales el propietario, no ofreció su consentimiento.

Con el éxito de Bitcoin aumentó la popularidad de la tecnología denominada Blockchain para una gran variedad de aplicaciones basadas en sistemas distribuidos, la cual permite el desarrollo de la Identidad Digital Autosoberana, Descentralizada o más conocida como Self-Sovereign Identity(SSI).

Este ecosistema, ofrece al usuario el control absoluto de su identidad digital, permitiendo compartir la información con quien y como quiera.

A pesar de los múltiples avances que se han realizado: normativas y estándares que facilitan su desarrollo e implementación a escala global, aun no se cuenta con una solución completa.

En principio, el objetivo de este trabajo es introducir al lector en la arquitectura, principios, elementos y funcionamiento desde un punto de vista teórico a fin de comprender las bases, filosofía e importancia de este ecosistema.

Una vez explicado lo anterior, se nombrarán los casos de uso llevados a cabo en Argentina, en particular el de DIDI, debido a la esperanza que la comunidad Iberoamericana de SSI tiene en él.

Luego, se realizará una PoC utilizando el proyecto Hyperledger Indy, el cual permitirá visualizar el comportamiento técnico de varios elementos.

Por último, se concluirá el trabajo exponiendo los desafíos, y las consideraciones actuales necesarias para que este modelo de identidad desplace al centralizado o federado que se encuentra implantado desde el comienzo de Internet.

1 Capítulo 1: Introducción a la Identidad Digital Descentralizada (SSI)

Este capítulo tiene como objetivo explicar teóricamente los elementos relevantes asociados a la Identidad Digital Descentralizada con el fin de que el lector tenga los conocimientos generales para seguir investigando este ecosistema emergente.

Se comienza definiendo el concepto de identidad digital, los modelos actuales, sus problemas y el esquema propuesto por SSI. De este último: la filosofía y los componentes participantes en la arquitectura.

1.1 Identidad digital

Antes de arrancar a hablar de SSI, me gustaría comenzar por definir la identidad digital, su esencia. Es importante entender este concepto, debido a que todos los que utilizamos alguna cuenta en cualquier servicio en Internet, indefectiblemente, poseemos una.

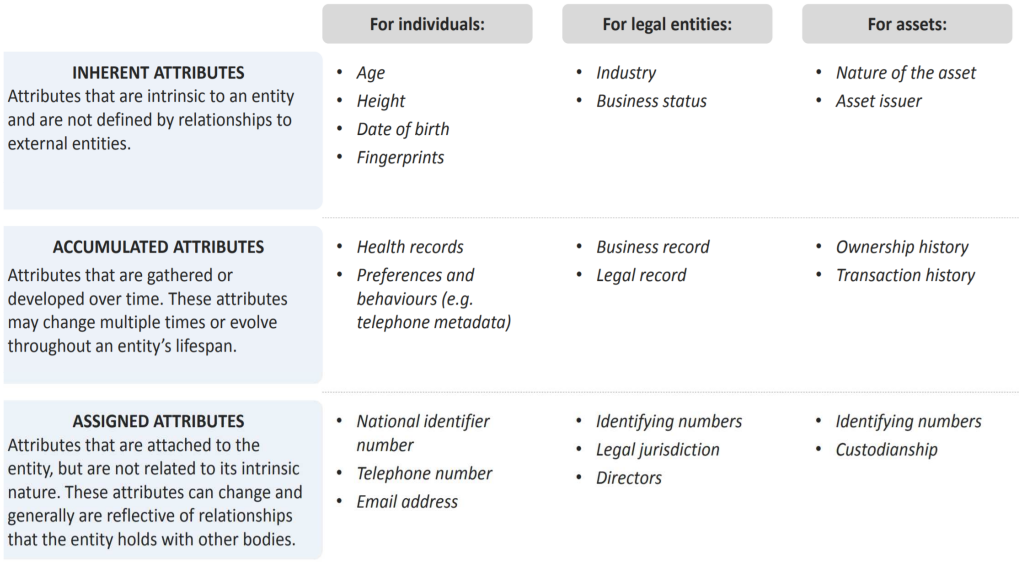
La identidad digital permite determinar de alguna manera quien o que se conecta a la red. Es decir, no solo las personas cuentan con una, sino también las organizaciones y activos que utilizan servicios a través de las telecomunicaciones.

Su definición no es tan trivial y de hecho se viene trabajando de manera notable desde el año 2005. El pionero en intentar abordarla es Kim Cameron al hablar de las leyes de la identidad [1]. Este arquitecto referente de Microsoft en dicha temática, resaltó la relevancia de la falta de una capa, la cual permita determinar quién o que se conecta a la red. El problema radica en que Internet fue pensado para conectar máquinas, no otra cosa.

Al poco tiempo de la publicación se empezó a realizar un taller para abordar, cada seis meses, cuestiones relacionadas a la capa faltante. Toda la tecnología y protocolos relacionados con la Identidad de Internet, nacieron o fueron tratados en este lugar, denominado IIW (Internet Identity Workshop) [2] y en la comunidad de Rebooting the Web of Trust (RWOT).

En párrafos anteriores dijimos que la identidad digital permite reconocer de cierta forma a algo en un servicio Web, pero aún no dijimos a través de cuales elementos. Para ello nos vamos a apoyar en la definición que Alex Preukschat utilizó en un meetup [3] en base a un informe elaborado por A

Blueprint for Digital Identity [4]. “Es una colección de atributos individuales que describen una entidad y determinan las transacciones en las que puede participar”.



	For individuals:	For legal entities:	For assets:
INHERENT ATTRIBUTES Attributes that are intrinsic to an entity and are not defined by relationships to external entities.	• *Age* • *Height* • *Date of birth* • *Fingerprints*	• *Industry* • *Business status*	• *Nature of the asset* • *Asset issuer*
ACCUMULATED ATTRIBUTES Attributes that are gathered or developed over time. These attributes may change multiple times or evolve throughout an entity’s lifespan.	• *Health records* • *Preferences and behaviours (e.g. telephone metadata)*	• *Business record* • *Legal record*	• *Ownership history* • *Transaction history*
ASSIGNED ATTRIBUTES Attributes that are attached to the entity, but are not related to its intrinsic nature. These attributes can change and generally are reflective of relationships that the entity holds with other bodies.	• *National identifier number* • *Telephone number* • *Email address*	• *Identifying numbers* • *Legal jurisdiction* • *Directors*	• *Identifying numbers* • *Custodianship*

Ilustración 1 Esquema de Identidad digital

Fuente: [4].

Como se observa en la imagen, los atributos difieren en los grupos de usuarios: individuos, entidades legales y bienes. Estos últimos, a pesar de la traducción, podrían ser seres vivos. Por ejemplo mascotas.

Si bien el conjunto total de atributos existentes es considerable, estos pueden clasificarse en tres grupos: inherentes, acumulados y asignados.

Los inherentes son aquellos intrínsecos a la entidad, por ejemplo la fecha de nacimiento de un individuo.

Los acumulados cambian a través de su vida. Ejemplo: historial médico de una persona.

Los últimos, además de no pertenecer a los inherentes, reflejan la relación del ente con otros cuerpos. Ejemplo: correo electrónico corporativo, el cual se asigna al empleado mientras ocupe un puesto en la organización.

1.2 Modelos de identidad

En este apartado, explicaremos los tres modelos de identidad digital: uno es el propuesto por SSI y los otros son los que actualmente se utilizan de forma cotidiana.

1.2.1 Modelo de Identidad Centralizada

Este es el más utilizado actualmente, son empleados por gobiernos o instituciones confiables como bancos y compañías de telecomunicaciones.

Se crea una identidad al registrar una cuenta (generalmente un nombre de usuario y contraseña) en un sitio web, servicio o aplicación.

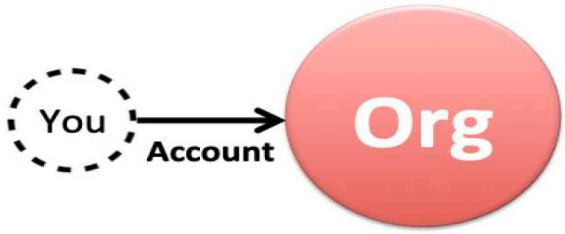


Ilustración 2 Modelo Centralizado

Fuente:[2]

En este diagrama, You(usted) es un círculo punteado porque solamente existe al crear una cuenta en un sistema centralizado. La organización es dueña de la misma, se la presta para que la pueda acceder a través de sus credenciales.

Algunos problemas relacionados con este enfoque:

- Si elimina todas sus cuentas en: sitios, servicios y aplicaciones; desaparecerá de Internet por completo.
- El trabajo de recordar y administrar todos los nombres de usuario y contraseñas; en algunos casos, otras herramientas de autenticación de múltiples factores, como las claves de un solo uso.
- Cada sitio aplica sus propias políticas de seguridad y privacidad. Por ejemplo, la complejidad de la contraseña.
- Ninguno de sus datos de identidad es portátil o reutilizable en

otro lugar.

- Las Bases de Datos utilizadas para gestionar la identidad centralizada, han sufrido las mayores violaciones de seguridad en la historia.

1.2.2 El modelo de Identidad Federado

Para solventar algunos de los inconvenientes mencionados en el modelo anterior surgió Identidad Federado. El cual inserta un intermediario, llamado Proveedor de Identidad o IDP, entre usted y la Organización.

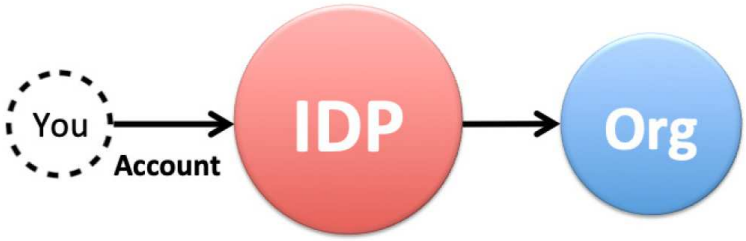


Ilustración 3 Modelo de Identidad Federado
Fuente: [2]

La cuenta se crea en el IDP, el cual inicia sesión y comparte datos con cualquier: sitio, servicio o aplicación que use ese proveedor. A este conjunto se le llama federación.

La Identidad Federada utiliza los protocolos de SAML, OAuth y OpenID Connect; lo cual permite hacer SSO (Single Sign-On). Es muy utilizada por redes sociales como Facebook o Google. También es una característica estándar de la mayoría de las intranets y extranets corporativas.

A pesar de todas las mejoras implementadas a este modelo desde el año 2005, aún no ha podido proporcionarnos la capa de identidad de Internet. Algunas razones son:

- No hay un IDP que funcione con todos los sitios, servicios y aplicaciones.
- Muchos usuarios se sienten incómodos al tener un intermediario en todas sus relaciones. Se recomienda ver un resumen de lo

sucedido entre Facebook y Cambridge Analytica [5] o el documental “Nada es privado” en Netflix.

- Si usted se desvincula de un IDP como Facebook o Twitter, todos esos inicios de sesión de cuenta se pierden.

1.2.3 El modelo de Identidad Descentralizado

La mejor analogía para explicar este modelo es exactamente la misma forma en la cual demostramos nuestra identidad el uno al otro: sacando nuestra billetera y mostrando las credenciales que hemos obtenido de otras partes confiables.

El proceso se realiza a través de los mismos elementos en formato electrónico. Existe una comunicación entre pares a través de una conexión punto a punto, en la cual cada uno es una entidad. En esta relación ninguno controla ni dispone de una cuenta en la otra, sino que se genera un vínculo directo entre ambos.

La característica de descentralizado se debe a que las redes punto a punto funcionan de dicha forma. Además el modelo utiliza tecnologías de conexión, las cuales no dependen de una autoridad central, como Blockchain.

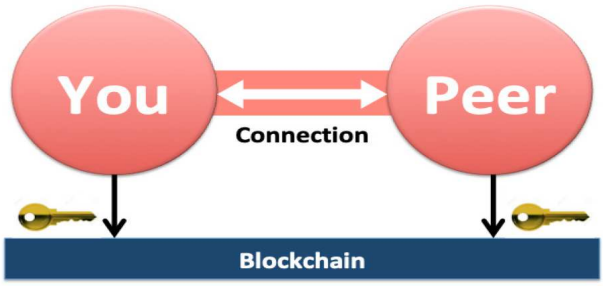


Ilustración 4 Modelo de Identidad Descentralizado

Fuente: [2]

El modelo fue inspirado por la tecnología Blockchain, la cual comenzó a surgir en 2015. A partir de allí hubo avances en criptografía y en nuevos estándares como Credenciales Verificables (VC) e identificadores descentralizados (DID). Estos son pilares fundamentales para la Identidad Digital Descentralizada.

La criptografía de clave pública es una forma de asegurar los datos a

través de algoritmos matemáticos. En lugar de usar Blockchain para criptomonedas, la gestión de identidad la utiliza para descentralizar la Infraestructura de clave pública (DPKI). En esencia, dicha tecnología y otras de red, denominadas DLT, pueden darnos una solución fuerte y sin dependencia de una autoridad central para:

1. Intercambiar claves públicas directamente para formar conexiones privadas y seguras entre dos pares.

2. Almacenar claves públicas en Blockchains para verificar las firmas almacenadas en las credenciales verificables, lo cual proporciona una prueba de identidad.

1.3 ¿Qué es SSI?

Mientras el modelo descentralizado comenzó a evolucionar, se lo apodó identidad autosoberana (SSI). Este término un poco confuso, creado por Christopher Allen, se debe a que no depende ni está sujeta a ningún otro poder u estado [2].

Es un concepto nuevo, el cual empezó a tomar relevancia en el 2016; es por ello que no existe un consenso en su definición. Sin embargo, podemos valernos de una propuesta por Alex Preukschat [3]: “Permitir que las personas controlen y administren su identidad digital para tener relaciones de confianza mientras se preserva la privacidad individual”.

Podemos decir que SSI tiene dos niveles: uno “filosófico” y otro tecnológico. El primero se basa en diez principios; el segundo se refiere a la tecnología, la cual utiliza criptografía y tecnologías descentralizadas, como Blockchain o DLT.

Si bien es una tecnología emergente y no se terminaron de establecer todas las cuestiones filosóficas de la identidad autosoberana, existe un consenso que abarca diez principios. Estos, deberían de guiar al ecosistema de SSI para generar confianza e interoperabilidad entre los servicios. Su autor, Christopher Allen, fue coautor del estándar de seguridad TLS. Él utilizó como fuentes el paper de Kim Cameron y la W3C Verifiable Credentials Working Group [6] para establecer sus principios en su artículo científico denominado “The Path to Self-Sovereing Identity” [7] .Los cuales son:

1. **Existencia**: la identidad digital debe ser gestionada por alguien que no existe solamente en el mundo virtual. Ejemplo: los certificados electrónicos, son administrados por personas. Según Casas, CEO en Validated ID, en una conferencia [8] “yo existo porque existo… no solo porque el señor banco me dice que existo…los prestadores de servicio me darán atributos vinculados a mi existencia que ya es previa”.
2. **Control**: los usuarios deben gestionar sus identidades, ser autosoberanos, poder decidir qué datos compartir, con quienes y por cuanto tiempo. También actualizarla, mostrarla u ocultarla.
3. **Acceso**: los propietarios gestionan sus propios datos sin

intermediarios, deberían poder recuperarlos fácilmente, al igual que todas las solicitudes dentro de su identidad. Un usuario no puede modificar todas las peticiones pero sí debe ser consciente de ellas.

4. **Transparencia**: sus sistemas y algoritmos deben ser: gratuitos, de código abierto, conocidos e independientes de cualquier arquitectura. Todos podrían examinar cómo funcionan, esto claramente incrementa la confianza de los usuarios.
5. **Persistencia**: las identidades deben durar el tiempo que el usuario lo desee, se tienen que actualizar en caso de ser necesario.
6. **Portabilidad**: los servicios y la información sobre la identidad deben ser transportables; además no tienen que mantenerse por una entidad externa. Esta podría desaparecer, el usuario cambiar de jurisdicción u otro motivo. Por lo tanto, se necesita garantizarle al individuo el control de la totalidad de sus datos.
7. **Interoperabilidad**: el objetivo es hacer que la información del usuario esté ampliamente disponible, incluso en otros países; crear identidades globales sin perder el control.
8. **Consentimiento**: el intercambio de datos solo debe realizarse con la aprobación del usuario para que estos sean válidos. Inclusive si un empleador, agencia de crédito o amigo presenten reclamos/solicitudes.
9. **Minimización**: ante un reclamo o solicitud se debe revelar solo los datos mínimos. Ejemplo: si solo se requiere una edad, se brinda ese dato y no la fecha de nacimiento. Este principio puede respaldarse con la divulgación selectiva.
10. **Protección**: el último principio corresponde a proteger los derechos de los usuarios, estos deben estar por encima de las necesidades de la red. Para garantizarlo, la autenticación de identidad tiene que realizarse a través de algoritmos independientes, ejecutados de manera descentralizada, resistentes a la fuerza y censura.

Como mencionamos anteriormente, SSI también tiene una parte tecnológica, comprendida de ocho elementos básicos:

1. Credenciales verificables o digitales
2. Emisores, titulares y verificadores.
3. Billeteras digitales.
4. Agentes digitales y centros.
5. Identificadores descentralizados (DID).
6. Blockchains.
7. Marcos de gobierno o de confianza.
8. DKMS (Sistema de Gestión de Claves Descentralizado).

1.4 Bloques principales de SSI

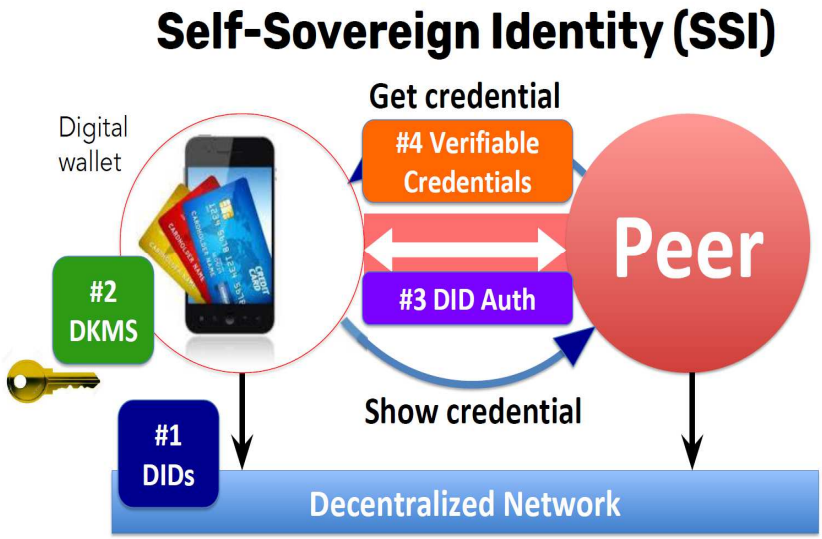


Ilustración 5 : Identidad Digital Descentralizada

Fuente: [9]

En la ilustración se visualiza a alto nivel la esencia de la identidad autosoberana, podemos observar los bloques, estándares y especificaciones mencionados en la página anterior.

Su principal característica es la comunicación entre dos pares, la misma está autenticada punto a punto, no existe un intermediario. En esta interacción, a alto nivel, simplemente se emiten y muestran credenciales.

Tal como en la vida diaria, podemos solicitar una licencia de conducir y luego mostrársela a otra entidad para que la verifique. Por ejemplo un oficial de tránsito, el cual confía en la autoridad emisora.

En la Identidad Digital Descentralizada, utilizaría un identificador único a través de la red descentralizada para constatar si la documentación es válida. Por lo tanto, podemos decir que se basa en una red de confianza.

Este modelo digital utiliza dispositivos informáticos como celulares y laptops, lo cual permite al usuario interactuar de forma directa y tener a su disposición las credenciales cuando se requiera.

Por último, en el caso que se extravíe el celular o esté comprometida alguna clave del identificador antes mencionado, entra en juego el DKMS (Sistema de Administración de Claves Descentralizada), el cual permite

recuperar la información asociada a dicho dispositivo sin necesitar de una autoridad central.

1.4.1 Credenciales verificables

Cotidianamente usamos credenciales verificables físicas: tarjetas de crédito, licencia de conducir, pasaportes, etc. No podemos vivir sin ellas porque nos proporcionan muchos beneficios; sin embargo, están expuestos a una serie de problemas. Los cuales pueden ser: extravío, robo, en el peor de los casos copiarse o clonarse sin que lo sepamos. Además para usarlos en línea necesita escribir todos sus detalles en un formulario Web, esto último consume mucho tiempo, es propenso a errores e invade la privacidad.

La idea de las credenciales verificables es convertir aquellas físicas en digitales y poder usarlas fácilmente en línea a través de dispositivos informáticos. Por ejemplo, en la pantalla del celular. Además de la facilidad de uso, ofrecen otras ventajas:

- No se pueden copiar ni clonar: deben demostrar que la credencial fue emitida para ellos y que además no está revocada. Se genera un link secreto, el cual solo es asociado al Holder (titular) de la credencial.
- Protegen más la privacidad: admiten divulgación selectiva, se puede presentar una prueba que solo demuestre lo mínimo y necesario. Ejemplo: en caso de necesitar saber si el sujeto es mayor de edad, se verifique si es verdadero o falso, evitando revelar la fecha de nacimiento y otros datos. Los cuales se mostrarían si se ofreciese el documento físico. Esto se conoce como ZKP (Prueba de Conocimiento Cero).
- Difíciles de robar: el atacante necesitaría el dispositivo y el método utilizado para autenticarse.
- Pueden delegarse: si el Holder lo requiere se puede autorizar a otra persona, emitiendo otra credencial. En la cual se coloca un atributo que indique el grado de parentesco.

Una credencial es cualquier conjunto de información que una autoridad confiable le emite a una entidad; este componente le permite convencer a otras que creen en la misma de que el elemento expedido es válido. Por ejemplo: un diploma otorgado por una universidad reconocida, certifica que un individuo posee un título académico.

Para calificar como credencial y por ende que exista esa relación de confianza, los atributos deben ser verificables de alguna forma. Necesitan criptografía e Internet. Este proceso puede realizarse en segundos y debe responder las siguientes preguntas:

1. ¿Se encuentra la credencial en un formato estándar para que pueda ser procesada por el verificador? ¿Contiene los datos que este necesita?
2. ¿Incluye una firma digital válida del emisor? (estableciendo así su origen y que no ha sido manipulada en el trayecto)
3. ¿La credencial sigue siendo válida? (no expiró o fue revocada)
4. ¿La credencial o su firma proporciona prueba criptográfica de que el titular es el sujeto de la misma?

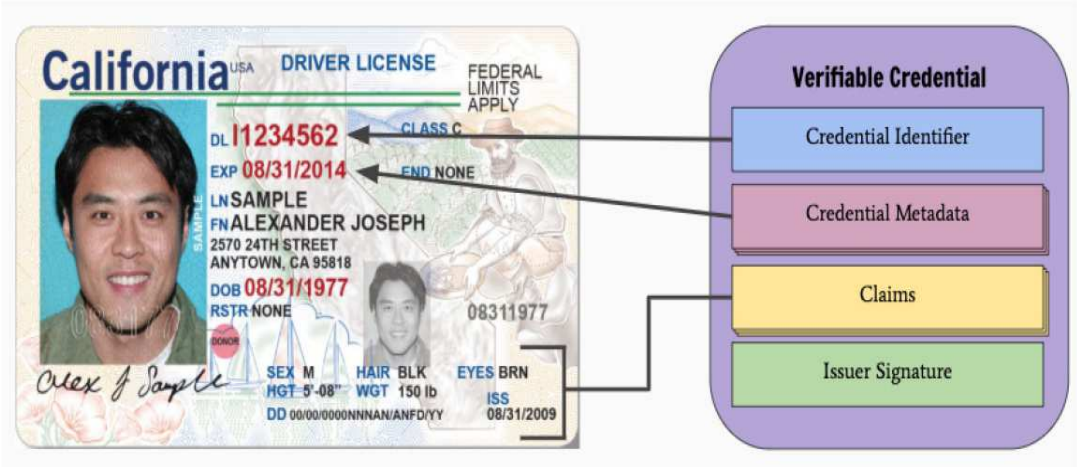


Ilustración 6 Mapeo de credencial física a 3 componentes de W3C

Fuente: [2]

La especificación del modelo 1.0 muestra cómo se puede estructurar una credencial digital para responder a las cuatro preguntas mencionadas

anteriormente.

La primera parte del paquete de credenciales verificables (Credential Identifier) corresponde a un identificador único, es el número mostrado en la parte superior de la figura.

La segunda parte (Credential Metadata) son los metadatos que describen la credencial. Por ejemplo, su fecha de vencimiento.

La tercera parte (Claims) hace mención a los atributos propios de la credencial. Por ejemplo: nombre, fecha de nacimiento, sexo, color de cabello, etc.

El contenido principal de una credencial verificable es al menos uno o un conjunto de reclamos/solicitudes (atributos). Este componente generado, de por si es un único Claim; por lo tanto, solo las declaraciones requeridas tienen que ser reveladas. Esta es una característica importante, la cual permite lograr un mayor nivel de privacidad y ofrece una gran ventaja con respecto a la documentación utilizada actualmente.

Otro componente, Issuer Signature, es la firma del emisor, este certifica la credencial verificable en la tercera parte, sirve como prueba criptográfica para demostrar quién lo expidió. Contiene: el tipo utilizado (como RsaSignature2018), su valor, fecha/hora de creación, el nonce y la clave pública del emisor.

Esta información es importante para que terceros puedan constatar los datos presentados. En caso de no conocer al ente confiable, puede optar por consultar un segundo triángulo de confianza. Lo cual se ve en la sección de Gobierno y en la capa cuatro de la Arquitectura de SSI.

Las credenciales además de expedirse, actualizarse y presentarse, se pueden revocar. A pesar de que el Holder posea total control sobre su uso, el emisor siempre tiene la opción de invalidarla en caso de no cumplir con los requisitos. Por ello la publica en una lista, ubicada en la capa uno de la Arquitectura de SSI (ledger generalmente), para que los verificadores logren chequearla cuando lo necesiten.

1.4.2 Emisores, titulares y verificadores

La ilustración muestra la terminología definida por el Grupo de Trabajo de Reclamaciones Verificables del W3C, para los tres roles principales involucrados en el intercambio de credenciales, cabe aclarar que una entidad puede tener distintos roles.

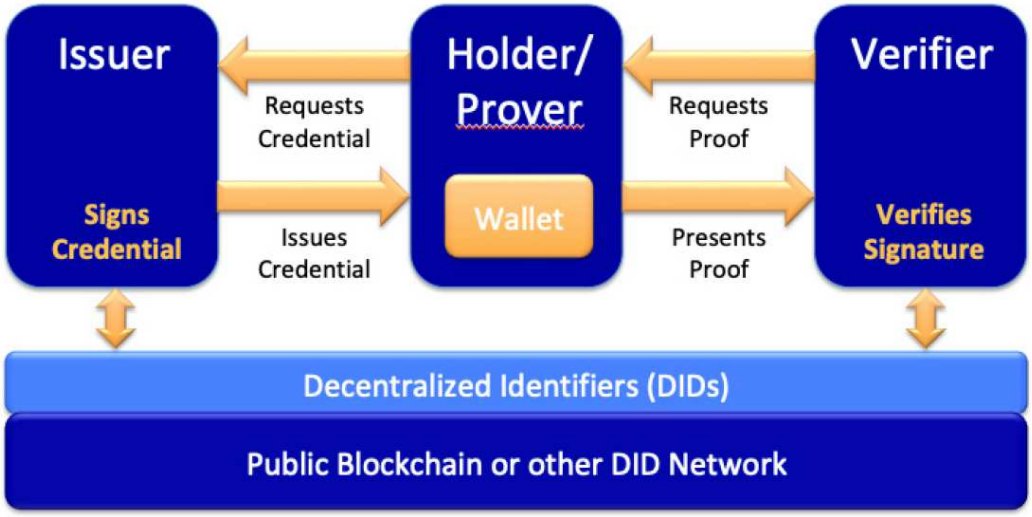


Ilustración 7 Roles en las credenciales verificables

Fuente [2]

1. **Los issuers** son la fuente de las credenciales, cada una tiene un emisor. La mayoría son organizaciones como agencias gubernamentales, instituciones financieras, universidades, etc. También pueden ser individuos y objetos cotidianos (IoT).

2. **Los titulares/comprobadores (Holder/prover)** solicitan credenciales a los emisores, las almacenan en la billetera (Wallet), y presentan pruebas o evidencias (Proof) cuando los verificadores lo soliciten. Pueden ser individuos, organizaciones (utilizan billeteras empresariales) u objetos cotidianos (IoT).

3. **Los verificadores (Verifier)** aquella persona, organización o cosa que busca la garantía de confianza sobre el sujeto de la credencial. Ellos solicitan pruebas al titular, si este acepta (tiene esa opción), el agente (software que administra la billetera) responderá con una evidencia, la cual puede ser comprobada.

El paso crítico en este proceso es la constatación de la firma digital del emisor, que generalmente se logra utilizando un DID (Identificador

descentralizado), lo cual se analizará con más detalle en este capítulo.

Imaginemos que un ex alumno (Holder/prover) de una universidad (issuer) desea obtener trabajo en una empresa (verifier) en el cual se requiere que el postulante tenga el título de Ingeniero.

En principio se establece una conexión punto a punto o P2P entre el ex alumno y el empleador, este último solicita pruebas (requests proof) de que el postulante es Ingeniero. Por lo cual, el egresado de nuestro ejemplo, necesita obtener la evidencia.

Posteriormente ocurre otra conexión punto a punto entre el postulante y la Universidad (Issuer) solicitando la prueba de la titulación (requests credential). La institución procede a emitir una evidencia firmada a través de su clave privada, la registra en la red descentralizada y se la entrega al ex alumno (issues credential).

Una vez obtenida la prueba, se la presenta al empleador (presents proof). La empresa verifica la firma de la universidad aplicando la clave pública de la institución contenida en un certificado digital emitido por una Autoridad Certificante; si concuerda puede aceptar la evidencia.

Cabe destacar que no existen entidades centrales y se establece un canal privado en cada conexión a través de un DID distinto. También, el ex alumno decide qué información revelar de su identidad y con quien compartirla, si la universidad no solicitó el promedio de notas, no es necesario presentarlo. A la idea de exponer lo mínimo se lo conoce como Pruebas de Conocimiento Cero (ZKP).

Por último, si el verifier hubiese requerido conocimientos de idiomas y el holder brinda su consentimiento para ello; puede entregar varias credenciales en forma conjunta.

1.4.3 Billeteras Digitales (Wallet)

Es aquella que permite almacenar las credenciales, protegerlas de robos y exposición. La cual opera con un agente digital (software) que es administrado por el usuario.

Una de las características es la portabilidad y estandarización. Las billeteras deben poder utilizarse en cualquier dispositivo y también sincronizarse. Además, es necesario que acepten cualquier credencial.

Otra particularidad es que permiten realizar una copia de seguridad, mover sus claves y credenciales entre billeteras digitales, incluso de diferentes proveedores; igual a como se utiliza en la vida cotidiana.

1.4.4 Agentes digitales y centros

Un agente digital es el software que opera la billetera. Tiene dos funciones: la primera permite a los propietarios administrarla y la segunda formar conexiones e intercambiar credenciales. Lo hacen a través del protocolo DID.

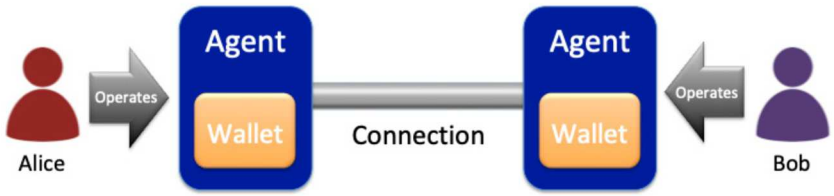


Ilustración 8 Comunicación entre agentes y billeteras de borde

Fuente: [2]

La imagen es una descripción a alto nivel de cómo los agentes y sus billeteras se comunican a través del ecosistema SSI.

Hay dos categorías generales de agentes: (según donde se encuentren) los de borde (Edge Agents) y aquellos que operan en la nube (Cloud Agents).

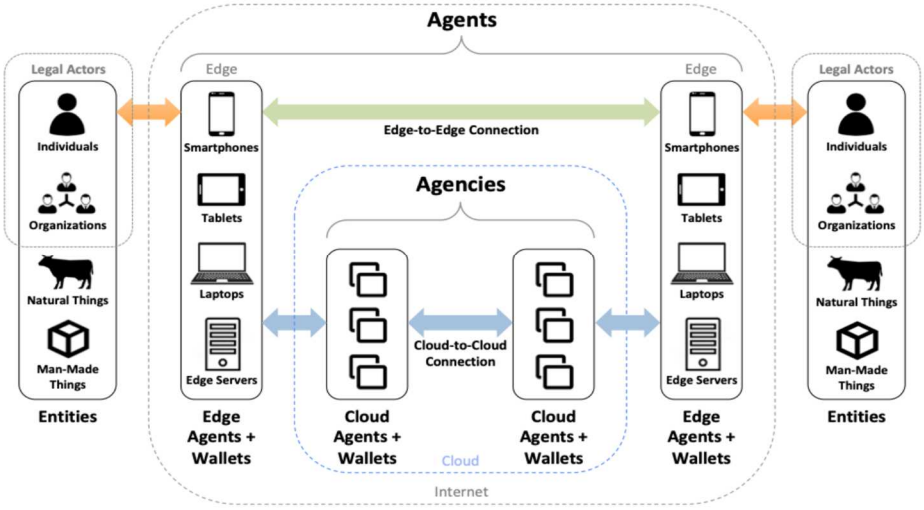


Ilustración 9 Diagrama de agentes y billeteras en SSI

Fuente: [2]

Los propietarios de identidades interactúan directamente con agentes perimetrales (en dispositivos locales) e indirectamente con agentes Cloud (que trabajan en la nube).

Los agentes ubicados en la nube pueden diseñarse para almacenar y sincronizar datos en nombre del propietario de la identidad. Por ejemplo: archivos, fotos, etc.

Los almacenes de datos cifrados se denominan centros de identidad/digital, los cuales procesan y mantienen información de las identidades, incluso después de la vida de un individuo. Esto podría ser de gran ayuda en la gestión y liquidación del patrimonio digital.

1.4.5 Identificadores descentralizados (DID)

Según Drummond Reed [10], coeditor de la especificación 1.0 del DID y experto en la temática, afirma que el Identificador Descentralizado y las Credenciales Verificables son los pilares de la arquitectura SSI.

Si bien la especificación 1.0 no es un estándar, está lo suficiente madura para diseñar sistemas y procesos.

Según la W3C DID Working Group [11] es un nuevo tipo de identificador, compuesto de un conjunto de strings de caracteres que identifican un recurso. El cual posee cuatro propiedades principales:

1. **Permanente**: nunca debe cambiar, independientemente si el propietario de la identidad usa diferentes dispositivos o proveedores de servicios.

2. **Resoluble**: deben poder recuperarse no solo las claves del propietario de la identidad, sino también las direcciones actuales para llegar a este o los agentes.

3. **Criptográficamente verificable**: el propietario de la identidad debe poder probar, utilizando criptografía, que tiene el control de la clave privada por cada llave pública asociada a ese identificador.

4. **Descentralizado**: este tipo de identificador, debe ser capaz de evitar puntos de falla mediante el uso de redes descentralizadas. Como: Blockchains, sistemas de archivos distribuidos, redes punto a punto, etc.

Las conexiones entre DID tienen ciertas propiedades: se interrumpen solo si una de ambas partes lo desean, funcionan a través de un canal privado, cifrado, firmado digitalmente y sin intermediarios.

Una entidad del mundo real puede tener muchos DIDs, de hecho, se recomienda uno por cada conexión, esto refuerza la privacidad.

Los pares de claves pueden generarse y gestionarse desde el lado del cliente, cada uno debe ser administrado por una entidad del mundo real, su controlador (controller)

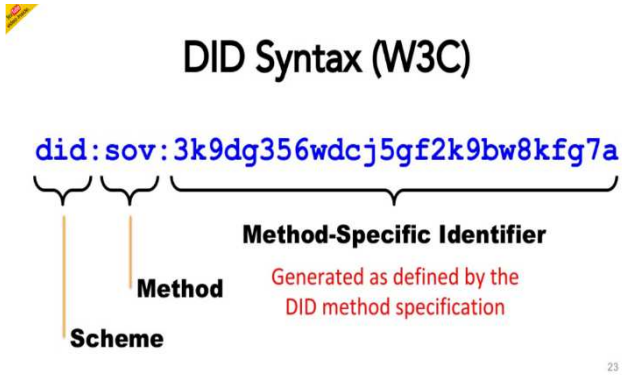


Ilustración 10 Ejemplo de un DID de Sovrin

Fuente: [12]

En la imagen, podemos observar el esquema (Scheme), método (Method) y su identificador (Method-Specific Identifier).

La primera parte es el esquema, el cual es un URN debido a que nunca cambia (a diferencia de una URL) y permite identificar objetos fuera de la web.

La segunda corresponde al método, este determina como el identificador (elemento siguiente) debe ser administrado. El ejemplo descrito corresponde a sov(Sovrin); la cual es una Blockchain pública diseñada específicamente y solo para SSI. [13]

La última parte describe al identificador, el cual es único dentro del contexto (ledger) especificado anteriormente.

Un DID funciona como un esquema de “clave-valor”, el primer campo corresponde al identificador y el segundo al documento asociado. No es necesario acceder a él totalmente, sino que se pueden referenciar fragmentos

o realizar consultas.

```
did:example:1234#keys-1
```

Ilustración 11 Ejemplo fragmento de DID

Fuente: [2]

Esta DID URL, identifica a una clave pública específica dentro del documento. Es similar al acceso de un fragmento en una página HTML.

```
did:example:1234?query=true
```

Ilustración 12 Ejemplo DID utilizando query

Fuente: Propia

1.4.5.1 Documento DID (DID Document)

No está pensado para visualizarse directamente en un navegador, sino que es utilizado o traducido por un software/hardware llamado resolver, el cual devuelve una estructura estándar de datos, un documento.

Es un archivo con formato JSON-LD y debe almacenarse en una fuente accesible, por ejemplo: una red Blockchain.

El documento contiene metadata sobre el sujeto (el recurso referido por el identificador) y una descripción sobre él.

Al que lo administra se le llama controller (controlador), el cual a veces puede no coincidir con el sujeto. Por ejemplo, cuando los padres tienen en su poder el documento que identifica a su hijo hasta que sea mayor de edad, recién allí, este último podría tener soberanía de su propia identidad digital.

Cada DID tiene asociado un documento, el cual nunca cambiará ni será reutilizado por alguien más [14].

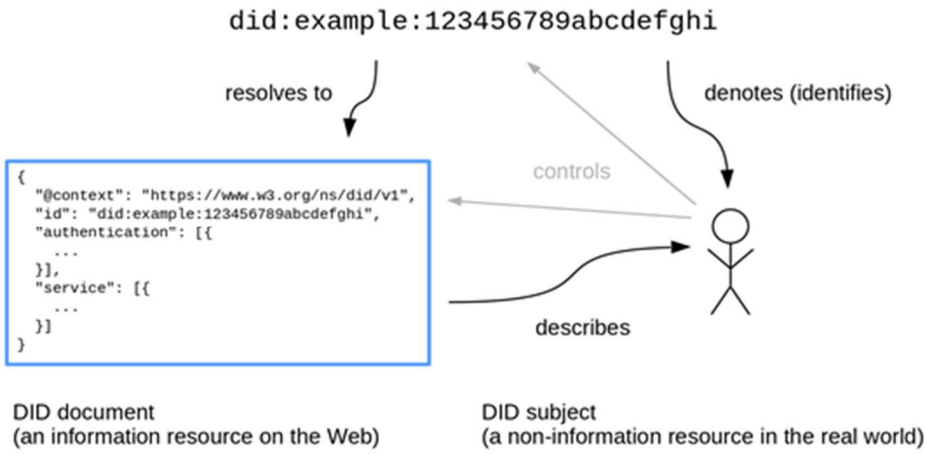


Ilustración 13 Relación entre DID, su documento y sujeto

Fuente: [2]

En la imagen visualizamos la relación entre el: DID, documento y sujeto, que en este caso, es también el controlador. El cual, administra los elementos mencionados.

El método es “example”, su identificador hace referencia al controller/subject y apunta a un documento (el cual describe al sujeto) obtenido en un proceso denominado **resolución**.

A pesar de que un documento puede contener cualquier información sobre el sujeto, por razones de privacidad, lo aconsejable es almacenar la metadata indispensable, con el fin de realizar una interacción confiable. Se recomienda:

- Utilizar una o más claves públicas u otros métodos de verificación, los cuales pueden ser utilizados para autenticar al sujeto en una conexión.
- Especificar los servicios que tiene permitido el sujeto en esa interacción, junto a los protocolos soportados. Por ejemplo: OpenID Connect (OIDC).
- Agregar metadata como: firmas digitales, pruebas criptográficas, etc.

```
{
  "@context": "https://www.w3.org/2019/did/v1",
  "id": "did:example:123456789abcdefghi",
  "authentication": [{
    // used to authenticate as did:...fghi
    "id": "did:example:123456789abcdefghi#keys-1",
    "type": "RsaVerificationKey2018",
    "controller": "did:example:123456789abcdefghi",
    "publicKeyPem": "-----BEGIN PUBLIC KEY...END PUBLIC KEY-----\r\n"
  }],
  "service": [{
    // used to retrieve Verifiable Credentials associated with the DID
    "id":"did:example:123456789abcdefghi#vcs",
    "type": "VerifiableCredentialService",
    "serviceEndpoint": "https://example.com/vc/"
  }]
}
```

Ilustración 14 Ejemplo de documento DID

Fuente: [6]

El formato del documento está definido en la especificación [6], siempre arranca con el atributo @context, este define la versión de configuración para que dos sistemas se puedan comunicar correctamente entre sí.

El atributo id, describe al sujeto, a través del método e identificador.

La autenticación es optativa, esta indica la forma en que ese DID va a ser verificado por su controlador.

El atributo type, dentro de authentication, especifica el método utilizado para autenticarse, en este ejemplo es RSA.

Por su parte, controller determina quién es el administrador del documento, en este caso es el sujeto, debido a que coincide con el campo id.

La clave pública se usa para firmas digitales, cifrado y otras operaciones criptográficas para permitir interacciones como autenticación. Además, son importantes para la autorización de las operaciones CRUD. Si un documento no contiene una, se debe suponer que el DID ha sido revocado y por ende no es válido.

El último atributo del documento define los endpoints de servicio, donde otros pueden interactuar con el DID.

Un endpoint de servicio podría ser un enlace, código QR o API para la interacción.

Este documento podría tener metadata. Por ejemplo: el timestamp (marca de tiempo); la cual registra su última fecha de: creación o actualización. Es muy útil para fines de auditoría.

Por último, pero no menos importante, se puede agregar una firma JSON-LD para verificar la integridad del documento.

1.4.5.2 Método DID (DID Method)

Es el mecanismo por el cual un DID y su documento son creados, leídos, actualizados y desactivados sobre una Blockchain específica o red [6].

```
did:sov:WRfXPg8dantKVubE3HX8pw
did:btcr:xz35-jzv2-qqs2-9wjt
did:v1:test:nym:3AEJTDMSxDDQpyUftjuoeZ2Bazp4Bswj1ce7FJGybCUu
did:ethr:0xE6Fe788d8ca214A080b0f6aC7F48480b2AEfa9a6
did:jolo:1fb352353ff51248c5104b407f9c04c3666627fcf5a167d693c9fc84b75964e2
```

Ilustración 15 Ejemplo de 5 métodos DID

Fuente: [2]

Hasta el momento se han registrado 40 en el registro de métodos DID [15], mantenido por el Grupo de Comunidad de Credenciales de la W3C. Cada uno debe definir lo siguiente:

- La sintaxis del identificador del método específico: largo de string, el cual se genera usando números aleatorios y funciones criptográficas, garantizando que este sea único.
- Las operaciones CRUD determinan como:
    - Create: registrar un DID en el ledger.
    - Read: leer y resolver un documento.
    - Update: actualizar un documento.
    - Desactive: desactivar un DID para no volver a utilizarse.
- Las consideraciones de seguridad y privacidad específicas aplicadas al método.

El estatuto del Grupo de Trabajo DID del W3C [11] incluye un documento para ayudar a evaluar que método es más eficiente a las necesidades de

un usuario o una comunidad en particular, ya que algunos pueden ser más adecuados dependiendo de su uso. Los cuales, entre otras cuestiones, difieren en escalabilidad y performance.

1.4.5.3 Resolución DID (DID resolution)

Es el proceso de obtener desde un DID, su documento asociado. Se puede usar para:

- Verificar la firma digital de un issuer (emisor) de una Credencial a través de su clave pública.
- Autenticar al controlador cuando se necesita logear en un sitio Web o una aplicación. Esto se logra a través de Did Auth.
- Para acceder a un servicio asociado al controlador. Ej: sitio web, red social, etc.

A continuación, se describe un escenario en el cual se utiliza la resolución.

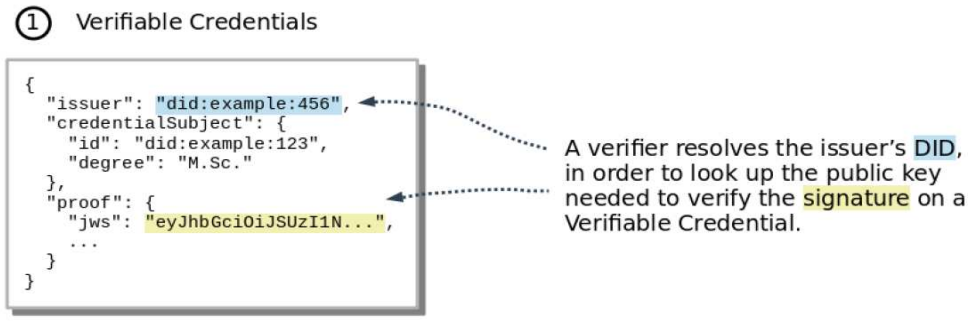


Ilustración 16 verificación de una firma en una Credencial
Fuente: [2]

El verificador quiere identificar al issuer de la Credencial, por lo tanto resuelve su DID, con el fin de obtener su clave pública y a través de la misma constatar la firma.

El proceso de resolución se basa directamente en una operación de lectura definida por el método. Puede variar considerablemente dependiendo de su diseño. Por lo tanto, en lugar de pensar en este proceso como un protocolo, debe considerarse una función abstracta o algoritmo que toma un DID (más parámetros opcionales) y devuelve el documento (con metadatos opcionales).

Los resolver están incluidos en todos los Sistemas Operativos modernos. También se puede implementar mediante una biblioteca nativa incluida en la aplicación o puede ser proporcionada por un tercero como un servicio alojado, respondiendo a las solicitudes de resolución a través de HTTP u otros protocolos (llamados enlaces).

Las formas mixtas también son posibles. No obstante, se recomienda que la resolución DID se integre directamente en una aplicación habilitada para ello, evitando la dependencia de terceros, lo cual podría introducir riesgos de centralización y seguridad [2].

1.4.5.4 Did Auth

Para establecer conexiones confiables, es crucial que los usuarios puedan demostrar que pueden controlar un DID específico [14].

Aún no es un estándar, ni siquiera una especificación, según Markus Sabadello, coeditor del paper “Introduction to DID Auth”, es una colección de arquitecturas y escenarios, los cuales permiten demostrar el control del identificador; una forma de especificar que somos nosotros o como menciona en su Webinar [16] “I am me” (yo soy yo).

A pesar de las diferentes variaciones, esta forma de autenticarnos(a un alto nivel) funciona a través de un ciclo challenge-response (desafío-respuesta), entre el Identity Owner (propietario) y Relying Party (parte confiable).

El primero tiene control del DID y debe probar a la parte confiable, que es quien dice ser.

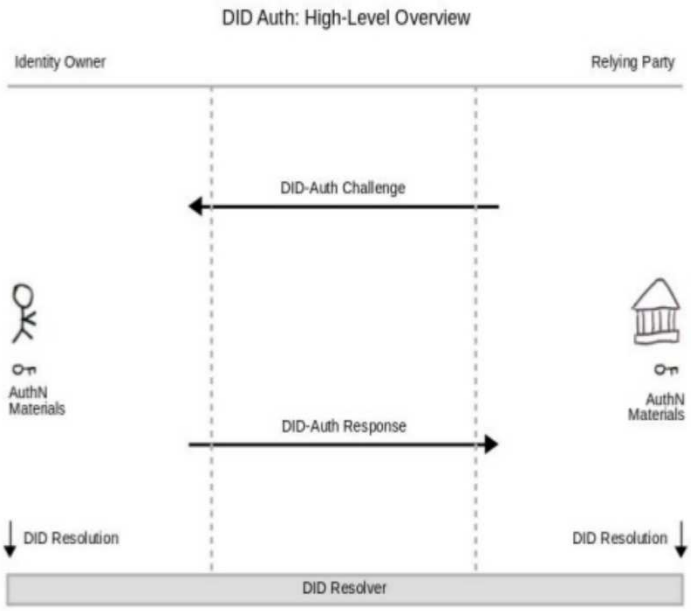


Ilustración 17 DID Auth a alto nivel
Fuente: [16]

En esta imagen se visualiza el challenge-response entre ambas partes.

Imaginemos que un usuario quiere logearse en un sitio Web. Este último solicita un Challenge criptográfico, el Identity Owner emite el Response (puede generar uno con su clave privada). La otra parte podría buscar la clave pública en el documento DID asociado al propietario y así constatar su identidad.

La interacción se puede dar a través de un navegador Web, una aplicación mobile, Bluethooth, QR, etc.

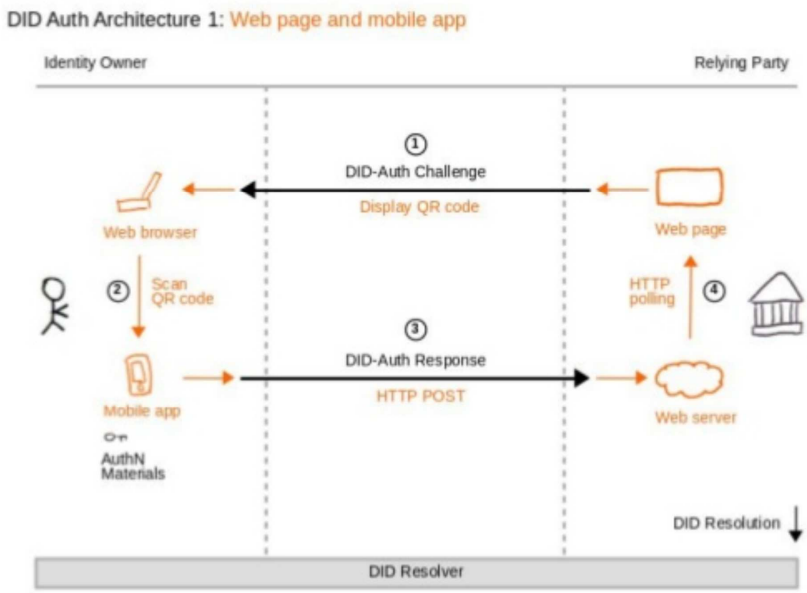


Ilustración 18 arquitectura 1 DID Auth: Página Web y aplicación móvil. Fuente: [16]

En la imagen se visualiza una arquitectura, en la cual el challenge y response son realizados por la Web page (página Web) y una Mobile app (aplicación móvil) respectivamente.

Nuevamente, imaginemos que un usuario quiere logearse en un sitio Web. El proceso sería el siguiente:

1) El sitio Web le solicita un Challenge, se visualiza el código QR en el Browser de la laptop.

2) El usuario escanea el código con una aplicación móvil a través de su agente de borde (Edge agent). Esta construye el response y se lo envía al Web server.

3) Este último realiza la resolución del DID, envía un HTTP polling, por ejemplo, uno de JavaScript a la página Web y se concreta el proceso de login.

En DID Auth, puede haber diferentes formatos: el JWT y el JSON-LD.

El primero es un documento creado específicamente para la autenticación y tiene muchas implementaciones. Es el más utilizado, codifica las notificaciones como un objeto JSON, puede ser firmado criptográficamente y transferido a otras partes.

El JSON-LD es especialmente adecuado para credenciales verificables ya que proporciona la misma funcionalidad de firma que un JWT.

Estos dos formatos de datos proporcionan la forma más sencilla y fluida de enviar la información requerida para DID Auth.

En SSI, la biometría no juega un papel importante pero se puede implementar. Por ejemplo, al querer desbloquear una billetera o un agente donde se almacenan las claves privadas y públicas para un DID.

Si la autenticación es biométrica, sus datos deben ser parte del documento DID para probar su propiedad.

En caso de que ocurra algún suceso como ha de ser un accidente, el documento debe ser actualizado para alinearse con el cambio biométrico.

1.4.6 Blockchains

La cadena de bloques permite a los usuarios almacenar datos de una forma inmutable, transparente y distribuida sin la necesidad de una autoridad central [14] .

Estas características se alinean a los principios de SSI, funcionando como una fuente de confianza; debido a que: ningún par tiene el control, no hay una autoridad central, son abiertas, transparentes y carecen de un único punto de falla o ataque.

En SSI se utiliza para registrar y resolver los DID a través de sus claves públicas; los identificadores permiten que las billeteras digitales y sus agentes se comuniquen e intercambien credenciales verificables de una forma criptográficamente segura a través de una red descentralizada.

Si bien existen diferencias entre los diversos tipos de Blockchains (sin permiso, híbrido, etc.) esto no resulta relevante porque un método DID puede admitir prácticamente cualquier red descentralizada.

Las cadenas de bloques resuelven un problema, el cual nunca tuvo solución en criptografía, porque son bases de datos distribuidas globalmente que sirven como fuente de confianza para claves públicas, sin estar sujeta a un único punto de falla o ataque.

1.4.7 Marcos de gobernanza

El objetivo final de toda la infraestructura es lograr un nivel de confianza aceptable entre las partes que interactúan.

La base de esta convicción es establecida por la credibilidad criptográfica, mediante el uso de DID conocidos, resolubles y de claves públicas en una red descentralizada.

No obstante, para que las entidades reales involucradas confíen en este ecosistema, se debe establecer un conjunto de reglas: comerciales, legales y técnicas conocidas como marco de gobernanza o confianza.

La autoridad de gobierno es la entidad que crea y administra dicho marco. Este define el conjunto de roles existentes, derechos, obligaciones y procesos transparentes para generar confianza en los usuarios del sistema.

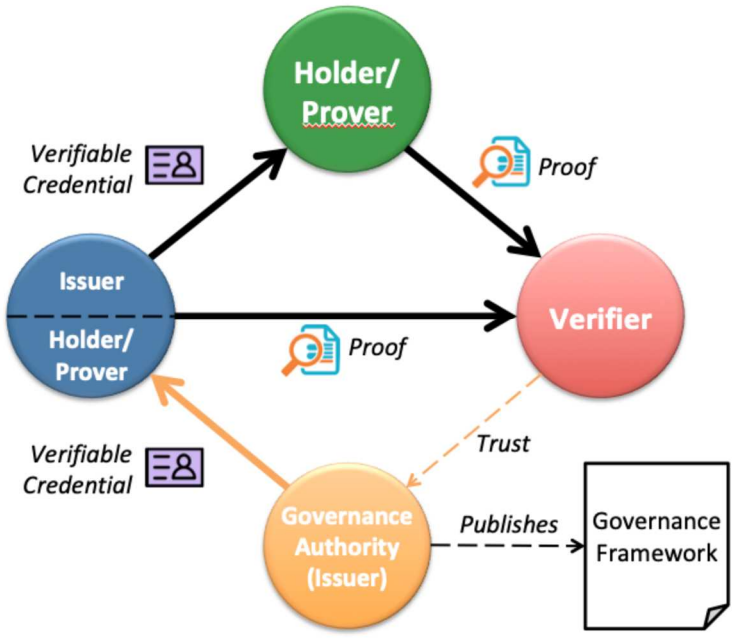


Ilustración 19 Segundo círculo de confianza

Fuente: [2]

En esta imagen podemos ver un segundo triángulo de confianza entre la Autoridad de Gobierno (Gobernance Authority), el Verifier y el Issuer, que ahora actúa como Holder/Prover.

Este tiene lugar cuando el Verifier no conoce al Issuer. Permite al

primero solicitar una prueba al emisor; la cual es una credencial autorizada por un marco de gobierno confiable por el verificador.

Los marcos de gobierno determinan las políticas y procedimientos a seguir cuando los emisores emiten una credencial. En algunos casos se especificarán términos y condiciones. Tales como: requisitos de seguro, variables legales, comerciales que los verificadores pueden tener en cuenta al momento de tomar decisiones.

Por último, es el secreto para que las credenciales verificables puedan escalar y funcionar en comunidades de cualquier tamaño. Desde una: ciudad, industria, nación o Internet en su conjunto.

1.4.8 Sistema Gestión de Claves Descentralizado (DKMS)

Así como los modelos tradicionales tienen una gestión de claves, el ecosistema SSI también; aunque este último es descentralizado.

A lo largo del trabajo se dijo que los DID no cambian; sin embargo las claves de adentro si pueden y de hecho es recomendable por motivos de seguridad.

También se mencionó que se posee uno por conexión, esto daría una cantidad enorme de identificadores para gestionar.

Entonces cabe preguntarse ¿qué pasa si perdemos el dispositivo que usa esos DIDs u ocurre algún problema con una clave? ¿cómo podemos gestionarlas?

DKMS es una forma estándar de administrar los DIDs, junto a sus claves públicas y privadas. Este grado de apertura e interoperabilidad, aumenta la usabilidad, seguridad y flexibilidad a los usuarios.

Es un estándar abierto y emergente, el cual tiene como objetivo estandarizar las billeteras para que los usuarios no tengan que preocuparse por la seguridad, privacidad o dependencia hacia un proveedor.

Este apartado dará una visión general sobre qué es y cómo se ve la arquitectura de DKMS.

Por último se explicará el proceso, a alto nivel, sobre cómo recuperar y revocar la clave.

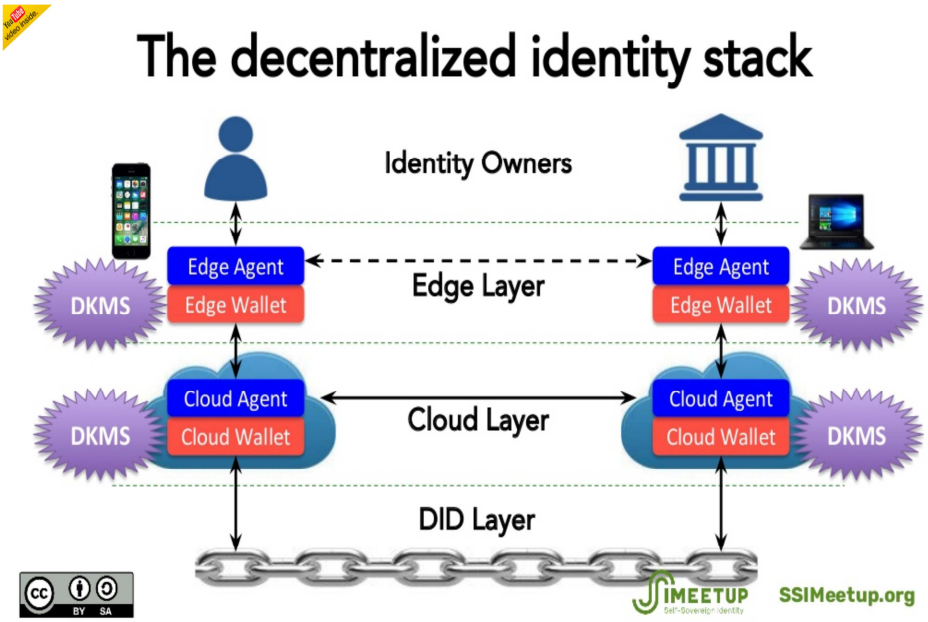


Ilustración 20 Arquitectura DKMS

Fuente: [9]

En la imagen se visualizan tres capas: DID, Nube (Cloud) y Borde (Edge).

La primera corresponde al ledger, donde se pueden gestionar los DIDs de forma descentralizada.

La capa de nube (Cloud Layer) permite realizar backup y restore de las billeteras ante algún incidente. También tener sincronismo entre los distintos dispositivos.

Por último, la capa de borde (Edge Layer) es aquella que interactúa directamente con el usuario, el cual controla la identidad.

DKMS, aplica a las capas Cloud y Edge debido a que, si se quiere obtener interoperabilidad entre las billeteras, ambas deben estar emparentadas.

En la capa de Borde, los Identity Owners necesitan usar dispositivos informáticos que estén en línea para usar SSI. Como un celular, laptop, etc.

Esta capa es la más segura para almacenar las claves privadas de los DIDs y administrar las identidades, ya que el dispositivo está bajo el control del usuario. También existe un menor riesgo de ataque en comparación a la nube.

Los dispositivos de Borde no siempre están en línea para interactuar con otros; además tienen un poder computacional reducido. Estos problemas se podrían resolver utilizando agentes en la nube (Cloud Agent) almacenando las billeteras en esta última capa.

Los agentes de nube permiten a los usuarios tener múltiples billeteras en diferentes dispositivos, lo que aumenta la flexibilidad. También se pueden utilizar para realizar copias de seguridad y recuperación ante la pérdida o pirateo de una de ellas.

El sincronismo se realiza con distintas claves por prevención; si se piratea una billetera, el atacante podría acceder a las restantes.

Debido a la importancia de los agentes (en ambas capas), hay que estandarizar lo máximo posible; permitiendo una mayor interoperabilidad en el ecosistema de SSI.

Recuperación de clave

Para llevar a cabo este proceso, los usuarios deben realizar periódicamente back ups de sus billeteras y almacenarlos cifrados en el agente ubicado en la nube o cualquier otro almacenamiento digital seguro.

La recuperación de la información debería ser trivial:

- Descargar una billetera.
- La conexión de los agentes (borde-billetera).
- Back up de una copia de seguridad de la billetera, hecha previamente.
- Descifrar a través de una llave especial, elaborada para tal fin.

Los mecanismos más utilizados para descifrar la clave son: recuperación fuera de línea y social. Ambos están basados en agentes ubicados en la nube.

El primero propone almacenar la clave en otro lugar, ya sea papel o hardware. Este soporte debe ser accesible y guardarse de forma segura.

La recuperación social permite repartir o fragmentar la clave de recuperación con terceros de confianza (trustees). Estos artefactos encriptados pueden almacenarse en las billeteras de los Owner.

El problema con este método es que los usuarios tienden a olvidarse quienes son los trustees. Ante ello, se podría pedir permanentemente a los usuarios que los elijan periódicamente.

Cabe destacar que la clave de recuperación es conocida por todos los dispositivos del usuario y para recuperarla se requiere un subconjunto de trustees. Los cuales deben aceptar ser parte del proceso.

Revocación de clave

Si una clave privada es conocida, se vuelve inútil, debido a que el verifier podría confiar en una identidad falsa. Por lo tanto, necesita revocarse y generarse una nueva.

Para generar transparencia y que los verifiers conozcan cuales son las claves confiables, se deben generar registros de revocación. Esta lista ofrece mayor seguridad a la conexión [14].

1.5 Arquitectura SSI

En este apartado se explicarán los objetivos generales de cada capa de la arquitectura SSI.

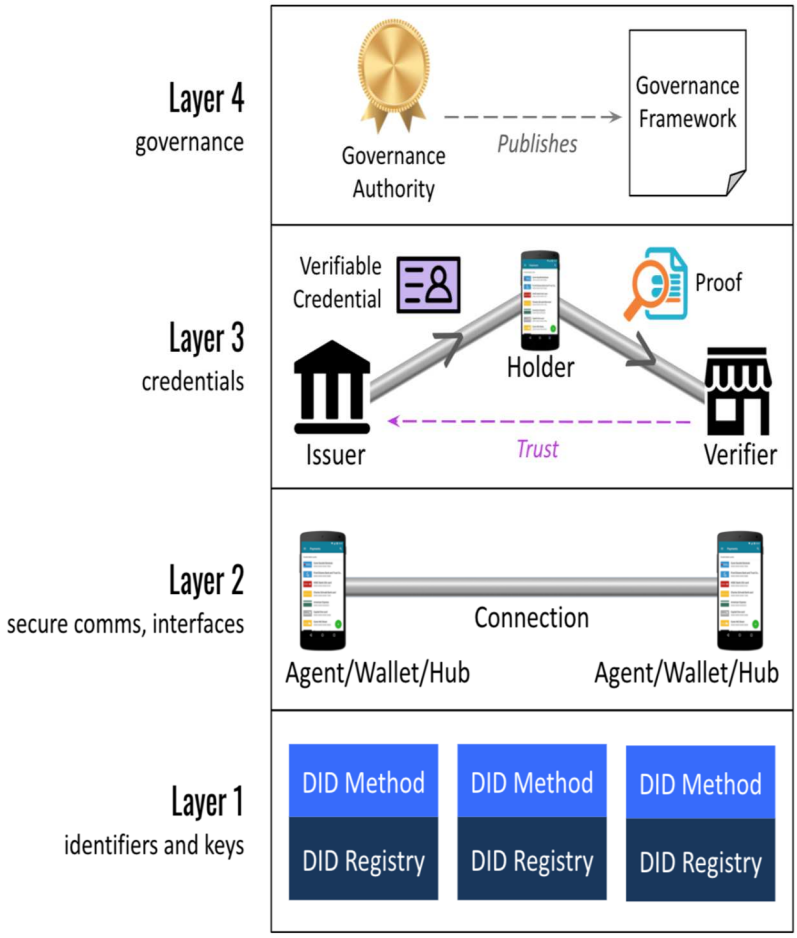


Ilustración 21 La pila de SSI de cuatro capas

Fuente: [2]

1.5.1 Capa 1: Claves e identificadores

Esta capa determina a qué hace referencia un identificador y cómo se podría probar su control utilizando criptografía. También debe permitir leer y escribir sin necesidad de autoridades centrales, una propiedad ampliamente mencionada en la comunidad Blockchain: "resistencia a la censura".

Existe un amplio consenso entre la comunidad SSI de que la mejor manera de proporcionar estas características es exponer un registro DID a través de alguno de sus métodos. Sin embargo, es útil realizarse algunas preguntas para elegir la tecnología correcta. Por ejemplo: ¿Cuál es la mejor

manera de implementar un sistema descentralizado? ¿Debe ser autorizado o sin permiso? ¿Cómo tiene que ser gobernado?

Existen distintas tecnologías, entre ellas: Blockchains para SSI, desarrolladas a partir de 2016, como Veres One e Hyperledger Indy. Las cuales tienen tipos de transacciones y registros que facilitan la administración de los DID. Además, se puede establecer el formato de la prueba de conocimiento cero (ZKP) compatible con el estándar W3C Verificable Credential Data Model 1.0.

1.5.2 Capa 2: Interfaces y comunicaciones seguras

Esta capa es donde los agentes digitales y billeteras participan a través de conexiones seguras (DID), las cuales utilizan la capa uno.

Aunque los actores de todo tipo (personas, organizaciones y cosas) están representados por estos agentes, billeteras y centros; la confianza establecida en esta capa es solo criptográfica. Se cree que:

- Un DID es controlado por otro par.
- Una conexión DID es segura.
- Un mensaje enviado en una comunicación es auténtico (no ha sido alterado).

Todas estas condiciones son necesarias pero insuficientes para establecer la confianza humana porque todavía no determinan nada sobre la persona, organización o cosa apuntada por el identificador. Por ejemplo, a una conexión DID no le importa si una parte remota es ética u honesta. Sin embargo, a las capas tres y cuatro, esto les resulta relevante.

Los problemas arquitectónicos se dividen en dos categorías principales**:** diseño de protocolo e interfaz.

Existen dos diseños de protocolos, uno basado en TLS y otro en mensajes usando DIDComm (Comunicación DID). El más utilizado es este último que a pesar de ser nuevo ya está ganando terreno en varias implementaciones. De hecho, se formó un Grupo de Trabajo (con el mismo nombre) en la Fundación de Identidad Descentralizada (DIF).

En este paradigma, las comunicaciones entre los agentes son similares al concepto de correo electrónico: es inherentemente asincrónico, permite

transmitir a múltiples partes al mismo tiempo, la entrega es a mejor esfuerzo y las respuestas pueden no llegar o hacerlo a través de diferentes canales.

Sin embargo, DIDComm tiene algunas diferencias con SMTP, entre ellas: los destinatarios se identifican por un DID en vez de una dirección de correo; toda la comunicación está asegurada (encriptada, firmada o ambas) por las claves asociadas a los identificadores descentralizados; los mensajes se pueden entregar a través de cualquier transporte: HTTP, Bluethooth, notificaciones push, códigos QR, etc.

Este protocolo puede usar fácilmente HTTP o HTTPS; sin embargo, ofrece flexibilidad para situaciones en las que solo una parte está conectada o donde los canales contienen muchos intermediarios no confiables.

El diseño de interfaz es la forma en que los programadores pueden utilizar las funcionalidades de la infraestructura SSI en aplicaciones o sistemas informáticos. Existen varios, entre ellos: el orientado a API, el cual consiste en dapps de billetera. Uport y Blockcerts son ejemplos de este tipo.

1.5.3 Capa 3: Credenciales

El objetivo de esta capa es hacer que las credenciales verificables sean interoperables. Para ello la comunidad se debe poner de acuerdo en el formato y protocolo de intercambio.

Lamentablemente, la comunidad tiene una divergencia significativa; con lo cual existe un riesgo de que esta capa no sea interoperable.

Para establecer un formato, la W3C publicó en 2019 el Modelo de datos de credenciales verificables 1.0**.** La cual utiliza la notación de objetos JavaScript para Datos vinculados (JSON-LD) para describir los diferentes esquemas y formatos de firma digital.

Se contemplan dos modelos: uno centrado en el intercambio simple de credenciales y otro que utiliza firmas criptográficas especializadas para facilitar las pruebas de conocimiento cero (ZKP). El último es el más recomendable, porque evita los problemas derivados de mostrar toda la credencial.

También existen diferencias en la comunidad al querer permitir, en ciertos casos, obtener información de una credencial desde otra fuente que

no sea el titular.

El enfoque de Hyperledger Indy, Aries y Sovrin fue poner foco en: la privacidad, la relación directa (nunca un intermediario) y la generación de pruebas de credenciales dinámicas (en el contexto de la conexión). Para ello, el intercambio se realiza con un protocolo P2P, utilizando agentes y DIDComm. Sus implementaciones ya están en producción.

Esta es la capa donde existe la mayor divergencia en arquitecturas y filosofías de protocolo. A pesar de que los esfuerzos de estandarización son activos, el consenso resulta difícil de alcanzar.

1.5.4 Capa 4: Gobierno

Como se explicó anteriormente, un Framework de Gobierno permite a los verificadores consultar información de un emisor en un segundo triángulo de confianza. Por ejemplo: ¿cómo saber si una licencia “x” fue emitida por una agencia real o confiable?

Es uno de los elementos más nuevos en el ecosistema. Si bien existen pocos de SSI, su trabajo inicial ha convencido a un grupo de arquitectos de que serán esenciales para alcanzar una amplia adopción del paradigma; porque son el puente entre la solución técnica y los requerimientos de: negocios, legales, sociales de dichas soluciones.

Para que dicho gobierno sea posible, existen Frameworks en cada capa.

2 Capítulo 2: Casos de Uso SSI

En este apartado, mencionaremos los casos de uso utilizados en Argentina. También se hará una prueba de concepto con el fin de comprender como se comunican tres entidades a través de credenciales verificables.

2.1 Casos de Uso en Argentina

Como se describe a lo largo de todo el trabajo, SSI es relativamente joven y se encuentra en constante cambio; antes de comenzar a mencionar algunos casos de uso cabe destacar que no todo lo que utiliza identidad digital descentralizada es considerado parte del movimiento. Para ello se deben respetar los principios propuestos por Kim Cameron.

En el Webinar 50 de SSIMeetup, Alex Preukschat define cuatro categorías de proyectos, de los cuales solo tres pertenecen al ecosistema. Estos son [3]:

1. Autenticación: aquellos que funcionan solamente como un "social login" de Facebook. Utilizan tecnología Blockchain o DLT en vez de un modelo centralizado/federado convencional. Fundamentalmente, no son consideradas propuestas de SSI debido a que no permiten la gestión de credenciales sin intermediarios.
2. Casos de uso SSI: son soluciones basadas en una plataforma. Las cuales aplican la identidad digital descentralizada para un uso concreto. Ejemplo: DIDI, Pasaporte Financiero.
3. Solución SSI: proyectos que utilizan alguna plataforma conocida del ecosistema con el fin de desarrollar una propia. A su vez, esta es utilizada por un caso de uso. Ejemplo: VidChain, Ubicua.
4. Plataforma SSI: aquella que aspira a ser un ecosistema completo. Ejemplo: Alastria Identity System-Alastria.Id.

País	Proyecto	Categoría
Brasil	Identity Tech	2 - Caso de Uso SSI
	BlockIoT	2 - Caso de Uso SSI
	eID+	1 - Autenticación
España	Alastria Identity System - Alastria.id	4 - Plataforma SSI
	VidChain	3 - Solución SSI
	Ubicua	3 - Solución SSI
	Qualif_ID	1 - Autenticación
	Niuron	1 - Autenticación
	Decode	Caso de Uso NO SSI
	Iden3 - Solución SSI	3 - Solución SSI
	Sikaris	3 - Solución SSI
	Esade	2 - Caso de Uso SSI
	Carlos III	2 - Caso de Uso SSI
Argentina	DIDI - Identidad Digital para la Inclusión	2 - Caso de Uso SSI
	Universidad Provincial sudoeste	2 - Caso de Uso SSI
	Pasaporte Financiero	2 - Caso de Uso SSI
Colombia	Universiad del Rosario	2 - Caso de Uso SSI
	Unal/Vivelab	Caso de Uso NO SSI
México	Tecnológico de Monterrey	2 - Caso de Uso SSI
Internacional	LAC-Chain	2 - Caso de Uso SSI

Ilustración 22 proyectos de países en Iberoamérica

Fuente: [3]

Como se observa en la imagen, solo España tiene proyectos en las cuatro categorías.

Argentina, posee tres proyectos de “Caso de Uso SSI”: Universidad Provincial sudoeste, Pasaporte Financiero y DIDI (Identidad Digital para la Inclusión).

El primero es un proyecto piloto realizado por la Universidad Provincial del Sudoeste. La cual utiliza Blockchain pública para emitir sus titulaciones oficiales [17].

Identidad digital, antes conocido como “Pasaporte Financiero”, es un caso de uso en desarrollo perteneciente a la empresa Signatura y elaborado en forma conjunta con RedLink durante el 2018. Su intención es realizar un proyecto de Identidad Digital Descentralizada eficiente para instituciones financieras. Ahorrando costos y tiempos gracias a la utilización de SSI [18].

El Proyecto DIDI es muy reconocido en la comunidad de SSI. Surge de la colaboración entre: ONG Bitcoin Argentina, laboratorio de innovación del Banco Interamericano de Desarrollo (IDB Labs), Accenture Argentina, RSK, NEC, Blockchain Federal Argentina, Atix Labs, Wicklow Capital y el Programa Semillas.

El objetivo principal es desarrollar una identidad digital privada que certifique los datos de las personas y valide las interacciones para permitir la inclusión a través de un sistema social/económico descentralizado.

El proyecto busca la inclusión de barrios vulnerables, ciudadanos que no pueden acceder al sistema financiero debido a que la mayoría son trabajadores informales y no poseen: títulos de propiedades, historial crediticio, registros oficiales de cursos y educación, etc. [19]

La idea es representar las interacciones formales e informales. Por ejemplo: probar lugar de residencia, acceder a servicios sanitarios con los que haya acuerdo, demostrar la experiencia laboral, etc.

En el barrio 31 se utilizan rondas, a través de un sistema de ahorro realizado por un grupo de personas conocidas entre sí. Ellos deciden cuánto dinero aportar y cada semana se sortea quién se llevará lo recaudado a modo de préstamo. El acceso al crédito se obtiene a partir de su participación; la cual puede visualizarse en la aplicación.

Se realizan programas de formación en alfabetización digital y gestión financiera debido a que la población con la que se trabaja no cuenta con dichas habilidades [20].

En principio, está a prueba en el barrio 31 aunque se planea que para el año 2022 esté completamente disponible. Su visión es utilizarlo en Latinoamérica, en lugares con características similares.

El proyecto se lanzará en la Blockchain RSK aunque la intención es que se pueda ejecutar en cualquiera.

El emisor de credenciales en principio es Semillas aunque la idea es que se amplíe a otros más [21].

2.2 Demostración de un escenario de SSI

El objetivo de este apartado es explicar los conceptos centrales de identidad digital a través de una prueba de concepto.

Se utilizará una práctica brindada en el curso: “Introducción a las tecnologías de Hyperledger con Blockchain” ofrecido por The Linux Fundation en el sitio Edx [22].

Esta propuesta es recomendada para quienes se inician en Identidad Digital Descentralizada e Hyperledger Indy. No obstante, es desaconsejable para desarrolladores debido a que están migrándose muchas funcionalidades al proyecto Aries, por ende el código podría variar considerablemente.

La demo permite visualizar a través de una interfaz Web: las interacciones, conexiones entre los agentes, el proceso para emitir y verificar credenciales.

La comunidad recomienda utilizar Docker en vez de una máquina local; debido a su facilidad de ejecución en un entorno virtual preconfigurado [23]. Poseer una cuenta allí es el único requisito previo y necesario para realizar la prueba de concepto sin problemas.

Alice, una ex alumna de Faber College, quiere ingresar a trabajar en Acme Corp. Esta compañía requiere profesionales en Marketing, por lo cual nuestra graduada necesita solicitar un certificado académico a la facultad y presentarlo como prueba en la empresa.

Primero se debe crear una nueva instancia en Docker. Para ello debe hacer click en el botón Agregar Nueva Instancia (Add new Instance).

Ilustración 23 Imagen entorno Docker

Fuente: [23]

Una vez iniciada sesión y creada la instancia se visualiza un entorno virtual igual al de la imagen. El cual nos proporciona una sesión máxima de cuatro horas, el tiempo necesario para trabajar sin apuro.

Notar que existe una Shell y un botón “OPEN PORT” (abrir puerto). Estas herramientas serán útiles para el desarrollo de la práctica.

En la Shell se deben escribir las instrucciones siguientes:

- **git clone https://github.com/hyperledger/education**
- **cd education/LFS171x/indy-material/nodejs**

Una vez ingresados los comandos para descargar el repositorio e ingresar al directorio de la demo, podemos comenzar a ejecutar la prueba. Para ello necesitamos escribir:

- **. /manage build**
- **. /manage up**

La primera instrucción compila los componentes y la segunda los ejecuta.

El Docker muestra una serie de puertos, los relevantes para nuestro ejemplo son:

- 3000: Alice
- 3002: Faber College
- 3003: Acme Corporation

Ilustración 24 Algunos puertos mostrados en el Docker
Fuente: [23]

Lo interesante es pensar que cada agente dentro del Docker no se ejecuta allí; sino en dispositivos y entornos diferentes. Por ejemplo: un celular o laptop.

Al hacer click en los puertos mencionados podemos notar que cada uno tiene características similares:

a) Una credencial del Gobierno con el nombre, email e id oficial de la Identidad.
b) Su DID generado y publicado (por estos) en el ledger de Hyperledger Indy.

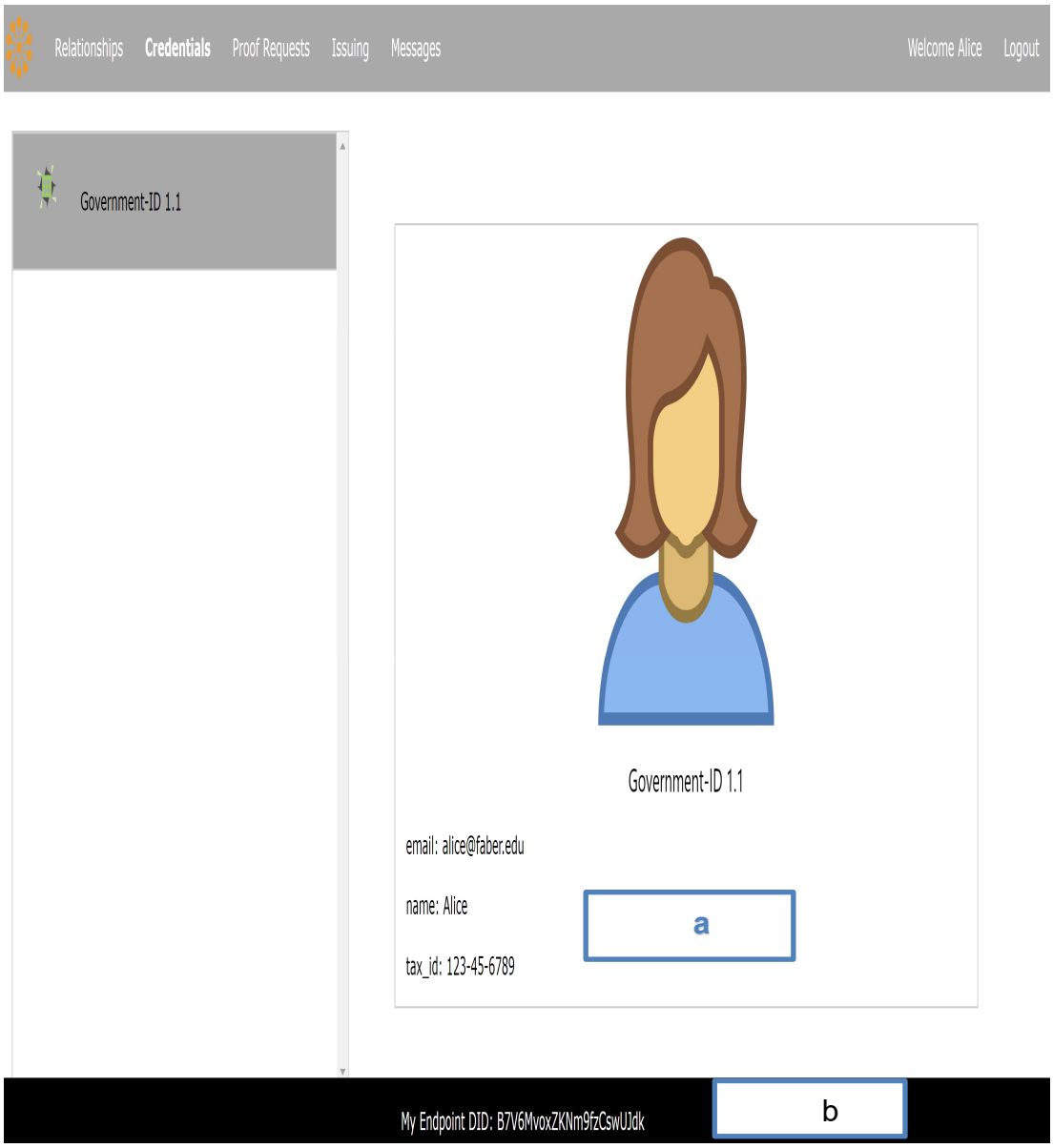


Ilustración 25 Interface del agente Alice

Fuente: [23]

En la imagen visualizamos los puntos a y b mencionados en el párrafo anterior.

Para que Alice pueda postularse en Acme Corp. necesita: conectarse/relacionarse con la universidad y esta compañía.

Una vez realizado este paso, debe solicitar una credencial verificable a Faber y por último presentarla en Acme Corp.

2.2.1 Estableciendo Relaciones entre Alice y Faber

Las relaciones se realizan a través de los DID. Alice envía una solicitud

de conexión utilizando el identificador del endpoint perteneciente a Faber.

En esta demostración, el DID de cada agente se muestra en la parte inferior de su ventana.

Como aclara el sitio donde se encuentra este ejemplo, la interfaz de usuario de una aplicación real será mucho mejor. Por ejemplo: Alice podría utilizar un código QR de Faber para iniciar sesión. Este método es utilizado por la demo IIWBook desarrollada por el equipo de VON en colaboración con Streetcred.id.

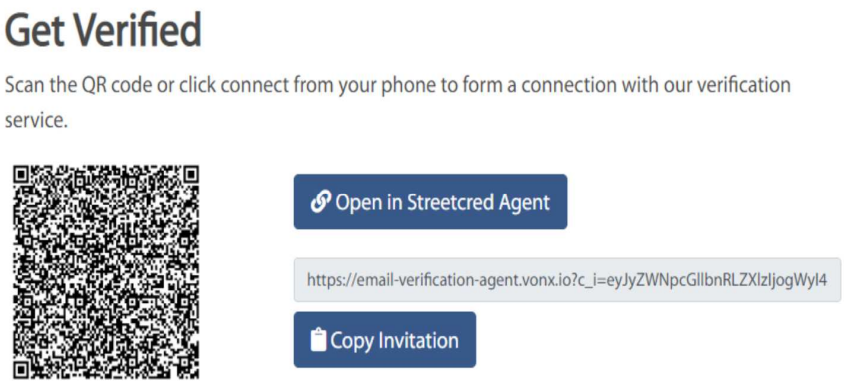


Ilustración 26 Ejemplo de conexión a través de QR
Fuente: [24]

Volviendo a la demo. Para crear una nueva relación entre Alice y Faber se necesita:

- Ir a la pestaña donde se encuentra el agente de la ex alumna. Haga click en Enviar nueva solicitud de conexión (Send New Connection Request).
- Copie el DID del Agente Faber. Regrese a la solapa anterior, péguelo en la ventana del punto de arriba y haga click en Enviar solicitud de conexión (Send Connection Request) .

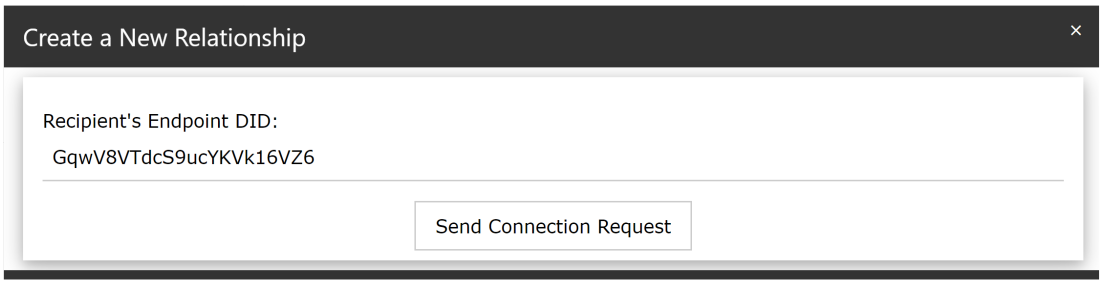


Ilustración 27 Crear una relación con Faber a través de su DID
Fuente: [23]

Una vez concretado lo anterior, se puede observar la conexión en la sección de Relaciones (Relationships) de ambos agentes.

Luego de establecerse el enlace entre DIDs, los dos deben intercambiar su credencial personal (otorgada por el Gobierno) para compartir su nombre.

Como este documento pertenece al titular, el agente le solicita que confirme si desea compartir la información.

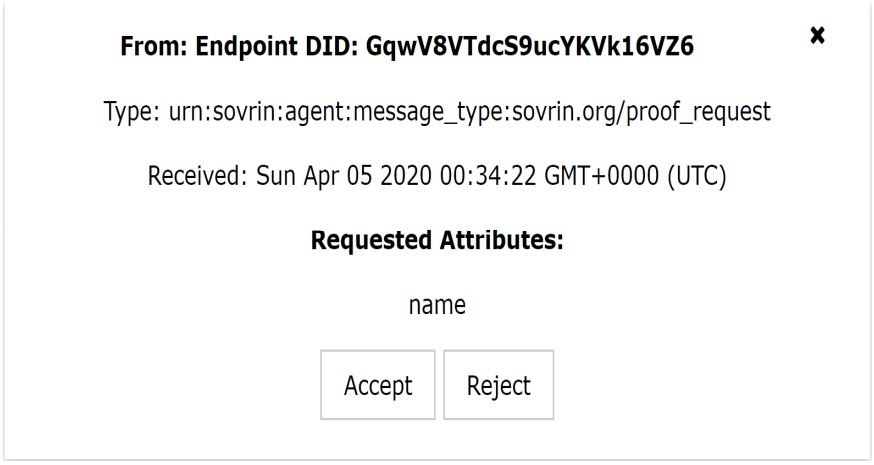


Ilustración 28 Solicitud de nombre del agente Faber a Alice
Fuente: [23]

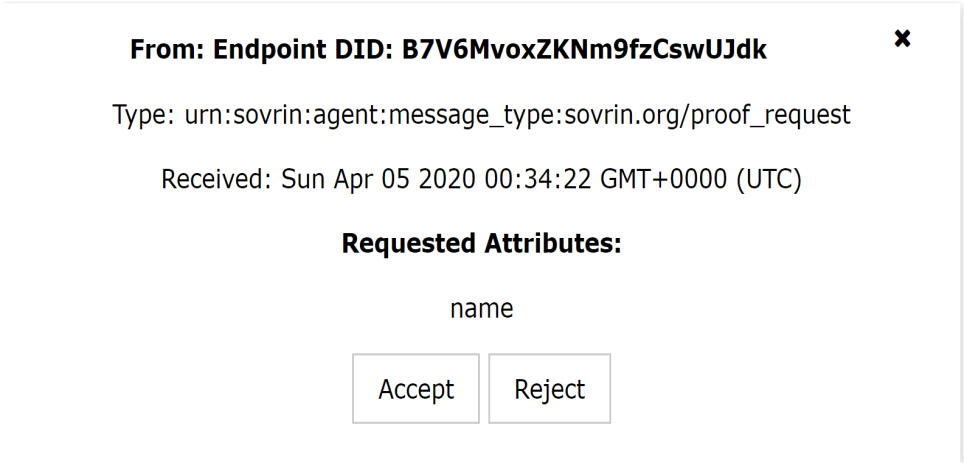


Ilustración 29 Solicitud de nombre del agente Alice a Faber

Fuente: [23]

Para seguir con nuestro ejemplo, ambos confirman la solicitud, reflejándose así los nombres de cada uno. Seleccione Aceptar (Accept) en las dos ventanas.

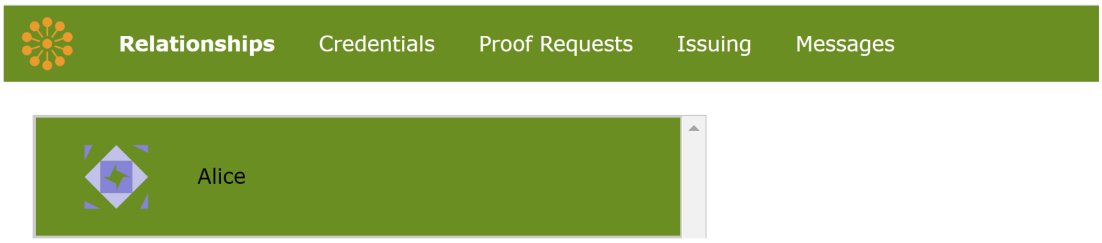


Ilustración 30 Sección relaciones de Faber con el nombre de Alice actualizado

Fuente: [23]

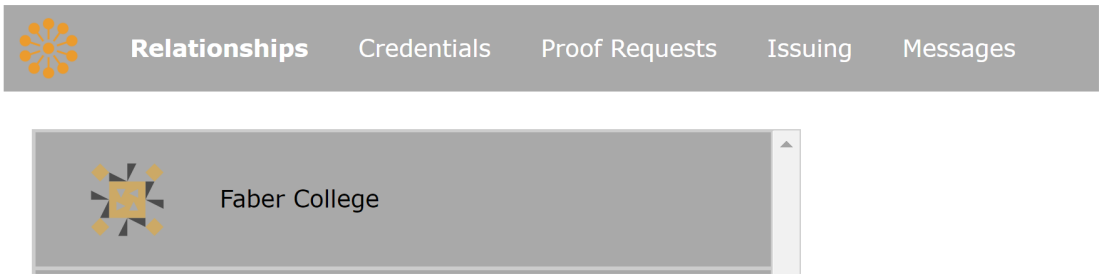


Ilustración 31 Sección relaciones de Alice con el nombre de Faber actualizado

Fuente: [23]

2.2.2 Emitiendo credencial de Faber a Alice

Antes de que la ex alumna solicite su credencial, es necesario para este

ejemplo **configurar Faber por única vez**. Para ello, hay que ir a su pestaña y hacer click en Emitir (Issuing). Aquí se generará:

- **Esquema de transcripción**: es una definición pública de qué campos (reclamos/claims) se utilizarán en las credenciales de Faber. Dado que no es privado, este puede ser emitido y utilizado por cualquier persona o institución. Ej: Universidad de Buenos Aires.
- **Definición de credencial**: un componente de Indy que vincula a un emisor (Faber), con un esquema de transcripción, registro de revocación(opcional), una serie de claves utilizadas para el cifrado y la firma de las solicitudes en las Credenciales.

Toda esa información es necesaria para que Faber pueda emitir una Credencial. Una vez explicado esto, procederemos a escribir un esquema y una definición de credencial en el ledger de Indy:

1. Ir a la pestaña del Agente Faber y hacer clic en Emitir (Issuing).

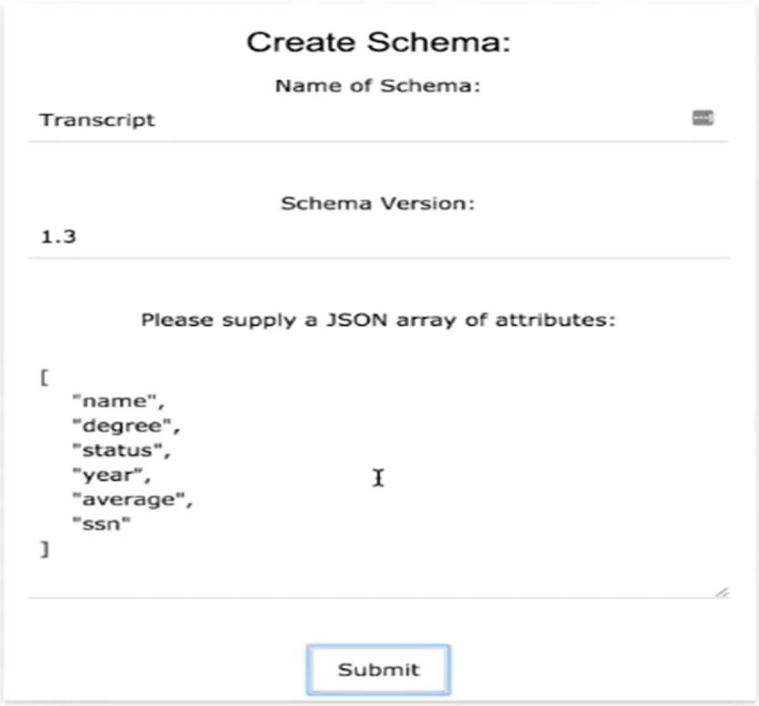


Ilustración 32 Ejemplo esquema de transcripción

Fuente: [23]

2. Escriba el nombre, versión y la lista de atributos del esquema. Nosotros dejaremos todo por defecto. Ejemplo: Transcript 1.3, nombre (name), título (degree), estado (status), año (year), promedio(average) y número de seguro social (ssn).
3. Hacer click en Enviar (Submit). Este evento publica el esquema en el Ledger de Indy para que Faber y otros emisores puedan utilizarlo.
4. En la ventana “Definición de credencial” (Create Credential Definition) seleccione el esquema recién creado (Transcript 1.3), elija un nombre, por ejemplo: Mytranscript y haga click en Enviar (Submit).

Ilustración 33 Ejemplo de definición de credencial

Fuente: [23]

El proceso de creación de una definición de credencial lleva un poco de tiempo porque Indy genera las claves criptográficas necesarias para la entidad. En este ejemplo se va a describir la emisión y recepción/aceptación:

1. En la ventana “Enviar Oferta de Credencial” (Send Credential Offer), seleccione Alice, la definición de credencial: MyTranscript y haga click en el botón.

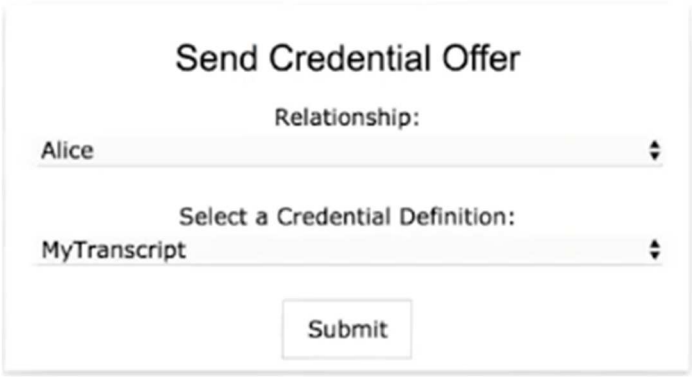


Ilustración 34 Emisión de credencial hacia Alice

Fuente: [23]

Una vez emitida la credencial, Alice la debe aceptar.

2. Para ello debe dirigirse a la solapa del agente de Alice y aceptar (Accept) el mensaje.

.

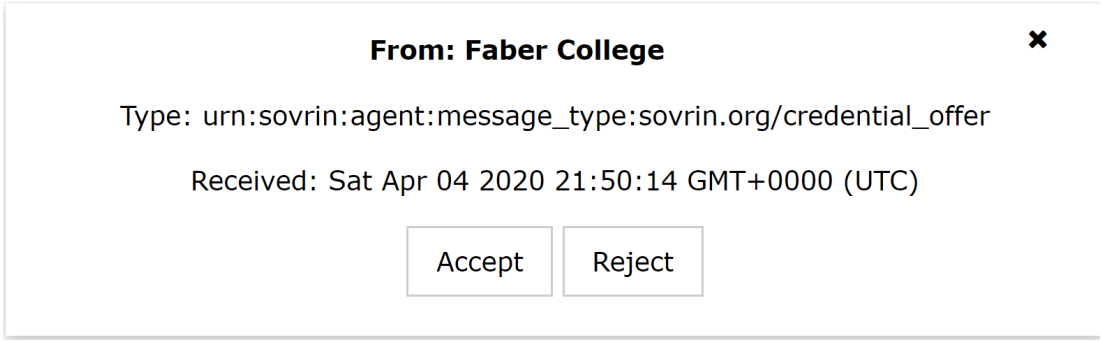


Ilustración 35 Mensaje de Ofrecimiento de Credencial de Faber a Alice

Fuente: [23]

Finalmente, Alice tiene la credencial académica emitida por Faber. Para visualizarla:

3. Ir a la sección de Credenciales (Credentials) y elegir Transcript 1.3.

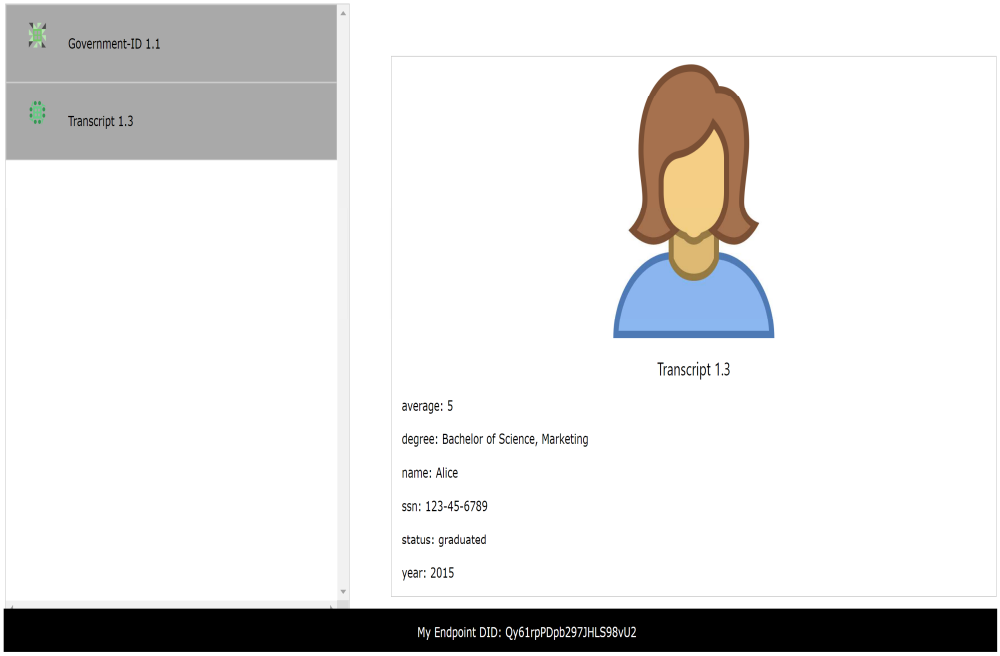


Ilustración 36 Credencial académica de Alice
Fuente: [23]

En el panel derecho podemos observar los datos de cada elemento definido en el esquema de transcripción. Por ejemplo, su promedio de notas (average): 5.

2.2.3 Solicitar empleo a Acme

Alice ha recibido su credencial académica, supongamos que se entrevistó con éxito, faltaría presentarle a Acme la prueba de que es graduada de Faber. Para ello, como se hizo anteriormente, primero debe establecerse una relación con la empresa:

1. Haga click en Enviar Solicitud de Conexión (Send New Connection Request).
2. Copie y pegue el DID de Endpoint de Acme. Luego haga click en Enviar Solicitud de Conexión (Send Connection Request).

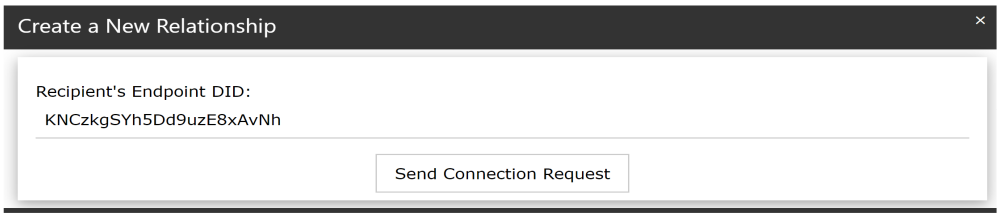


Ilustración 37 Solicitud de conexión de Alice a Acme

Fuente: [23]

Una vez realizados los dos pasos, se establece la conexión. Al igual que Alice-Faber, ambos se envían una Solicitud de Prueba para obtener información de la credencial del gobierno que cada uno posee.

Los dos revisan sus mensajes y observan que tienen una solicitud de prueba del atributo nombre (name) requerido por la otra parte.

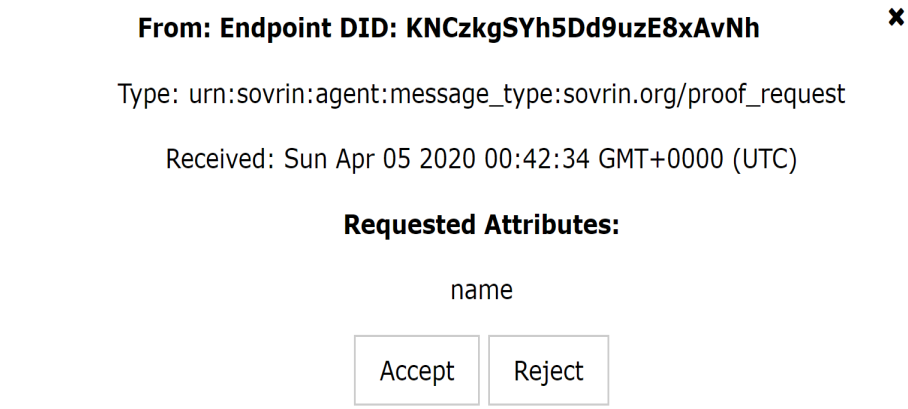


Ilustración 38 Solicitud prueba de nombre de Acme para Alice

Fuente: [23]

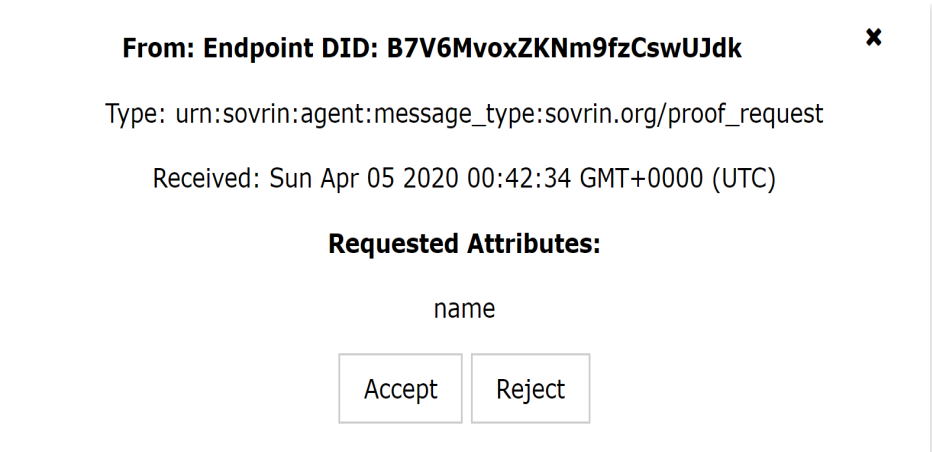


Ilustración 39 Solicitud prueba de nombre de Alice para Acme

Fuente: [23]

Una vez aceptadas las solicitudes, Acme envía otra con el fin de obtener la credencial de Alice. Aquí tenemos que usar un poco de “pirateo” para que funcione esta demostración, no es así como va a utilizarse en una aplicación real, pero es suficiente para transmitir la idea.

Desafortunadamente en esta práctica, la empresa no conoce la información para construir la Solicitud de prueba de las credenciales; debe obtenerla de Faber.

A fines prácticos, se va a hacer uso de “copy-paste” (existen otras formas en una aplicación real). Para ello:

1. En la solapa de Faber, haga click en Solicitudes de Prueba (Proof Requests). Deje a Alice en Relación (Relationship), elija Datos de Transcripción (Transcript-Data) y apriete en Enviar.
    a) Se mostrará en el cuadro de la derecha un texto (estructura JSON) con los campos "nombre": "Datos de Transcripción".
    b) Copie todo el texto.

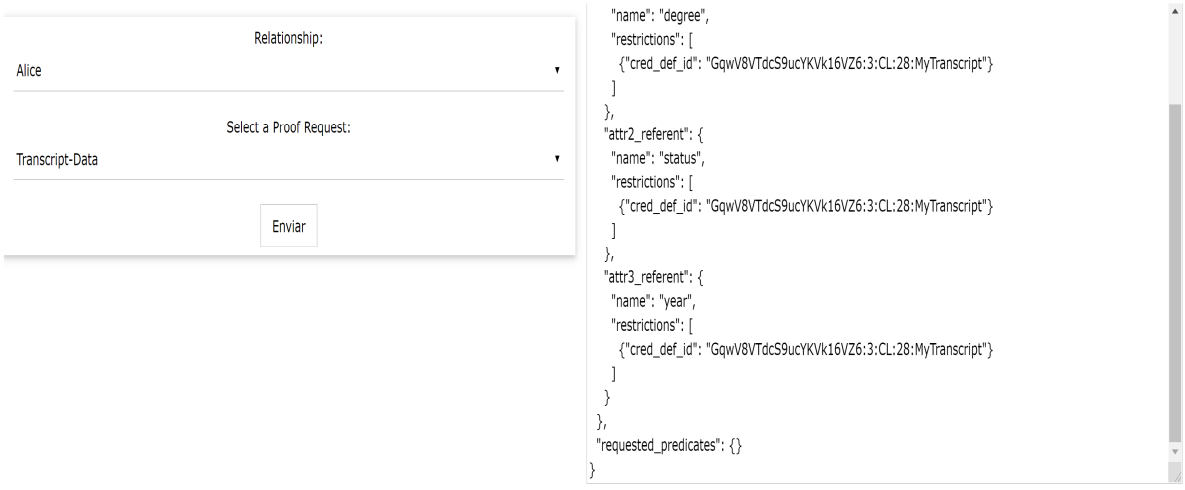


Ilustración 40 Copiando Solicitud de Prueba de credenciales Faber-Acme
Fuente: [23]

2. Vaya a Acme para solicitar la Prueba de credencial. Realice lo siguiente:
    a) Haga click en Solicitudes de Prueba (Proof

Requests).

b) Seleccione la relación (Relationship), en este caso, Alice.

c) En el campo de abajo elija: Otro ->Pegue la prueba aquí (Other-> Paste Proof Request Here).

d) Pegue el texto copiado de la pestaña Faber en el cuadro de la derecha.

e) Click en Enviar (Submit). Aquí se manda la solicitud de Prueba a Alice.

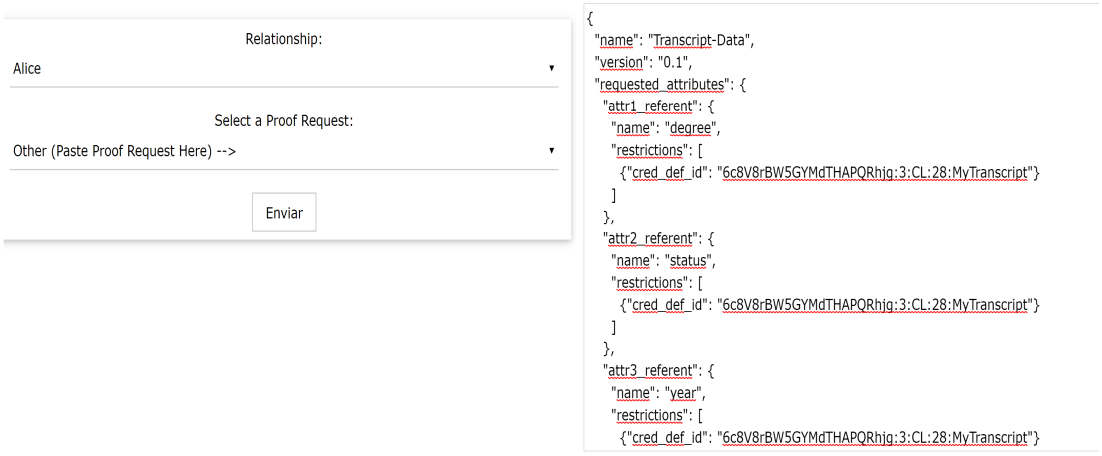


Ilustración 41 Solicitud de prueba de Acme a Alice a través de método manual

Fuente: [23]

3. Vaya al agente de Alice y haga click en Mensajes (Messages) y por último en Aceptar (Accept).

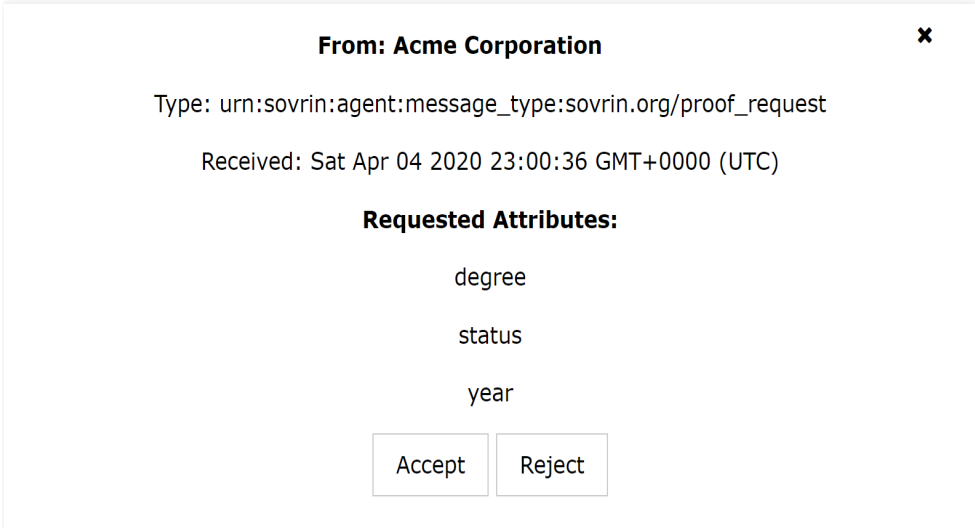


Ilustración 42 Mensaje de solicitud de prueba de Acme a Alice

Fuente: [23]

En la imagen observamos que Acme solicita prueba de: grado, estado y año.

Una vez que Alice aceptó, Acme puede ver en su apartado de Relaciones (Relationships) las pruebas solicitadas a la actora de nuestro ejemplo.

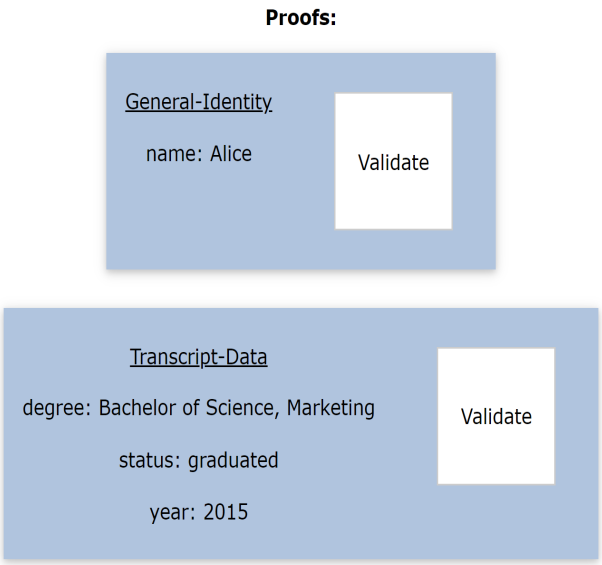


Ilustración 43 Pruebas de Alice en Acme

Fuente: [23]

Acme puede verificar que: los reclamos vinieron de Faber, fueron

proporcionados por Alice y no fueron alterados. Una vez validada la información, en nuestro ejemplo optimista, ella consigue el trabajo.

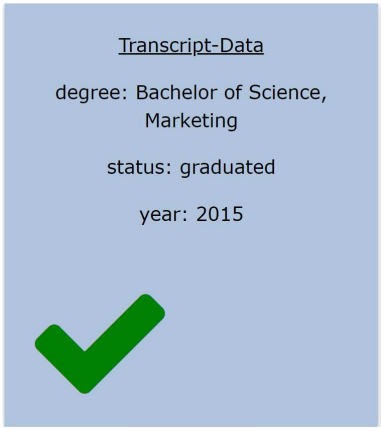


Ilustración 44 Acme valida prueba proporcionada por Alice
Fuente: [23]

3 Capítulo 3: Desafíos de SSI

A pesar de los beneficios y avances constantes, existen varios obstáculos para su adopción a gran escala.

Este capítulo describirá los inconvenientes principales tanto a nivel global como en Iberoamérica.

En un evento reciente llamado "El futuro de la identidad digital" presentado por Citi Venture se señaló que había tres desafíos principales para la implementación de SSI [25]:

- Construyendo el nuevo ecosistema SSI
- Gestión de claves descentralizada
- Acceso sin conexión

3.1 Construyendo el nuevo ecosistema SSI

Este enfoque es un cambio de paradigma de la identidad digital; por lo tanto, recién se verá si es viable o no a medida que las industrias, gobiernos y otros organismos comiencen a aceptarse mutuamente bajo este esquema.

La infraestructura aún no está madura, hay mucho trabajo por hacerse; aún falta interoperabilidad y portabilidad entre los distintos componentes de la arquitectura, tales como: agentes, DIDs, formato de credenciales, enfoques de gobernanza, etc.

3.2 Gestión descentralizada de claves (DKMS)

Como se explicó anteriormente, el extravío de una clave de una billetera podría equivaler a la pérdida completa de la identidad digital. Si bien existen mecanismos para gestionar las llaves y estandarizar wallets, aún queda mucho trabajo por delante.

Según Drummond Reed [2] la gestión de llaves siempre fue el talón de Aquiles en la adopción de la criptografía y la infraestructura de clave pública (PKI). Muchos expertos creen que solo puede ser administrado por grandes empresas y proveedores de servicios centralizados: bancos o agencias gubernamentales. El autor lo considera como uno de los principales obstáculos a superar para que SSI tenga éxito en el mercado.

3.3 Acceso sin conexión

Se necesita determinar cómo se usará la identidad digital en caso de la falta de acceso a Internet. Por ejemplo, un policía podría necesitar verificar la licencia de conducir en un lugar donde el acceso simplemente no esté disponible. Los proyectos SSI deberían poder funcionar fuera de línea o con conectividad intermitente; no obstante, este desafío se encuentra lejos de ser resuelto.

Por su parte, Alex Preukschat describió desafíos complementarios a los mencionados. Los últimos dos corresponden solo a Iberoamérica, debido a la falta de recursos en comparación a los países más desarrollados [26].

- La desconexión del mundo Blockchain con el de SSI.
- La innovación tecnológica como marketing.
- La falta de una red de apoyo para financiar el emprendimiento en el mundo SSI.

3.4 La desconexión del mundo Blockchain del mundo SSI

Según el autor, existe poca comunicación entre las comunidades Blockchain y SSI; a excepción de pocos proyectos como Uport o DIDI.

El primero, es el proyecto más activo a nivel global del mundo Blockchain.

DIDI por su parte, es la referencia local del mundo de la ONG Bitcoin Argentina. A pesar de ser una buena iniciativa se espera un despliegue más amplio de recursos.

Es necesaria una participación activa en: los principales foros de SSI, grupos de trabajo de identificadores descentralizados (Decentralized Identifier-DID) que operan en el W3C, comunidad Decentralized Identity Foundation (DIF), diferentes proyectos de Hyperledger Indy y en el trabajo realizado por el equipo de MyData. También de otras iniciativas que están en proceso de fomentar la interoperabilidad que aún no se han hecho públicas.

3.5 La innovación tecnológica como marketing

Según Preukschat, en Iberoamérica (más que en los países industrializados) no se aplican los recursos requeridos para generar avances significativos en el paradigma. Por el contrario, son gestionados con el fin de crear una imagen de: vanguardia y progreso; sin intencionalidad de innovar realmente.

3.6 La falta de una red para financiar el emprendimiento en el mundo SSI

Para que el mismo sea más sólido es necesario que otros emprendedores y empresas exitosas inviertan permanentemente en la siguiente generación. Además, los países deben crear entornos favorables para atraer talento y capital.

Además de estos desafíos, Daniel Hardman propone el siguiente [27]:

- **Facilidad de uso**: Los proyectos deberían utilizarse de forma intuitiva. Actualmente, los usuarios están acostumbrados a tomar pocas decisiones sobre el manejo de su identidad [28].

4 Conclusión

A lo largo del trabajo se ha descrito el concepto de identidad digital, modelos que la tratan y un nuevo paradigma que está empezando a tomar peso, principalmente, por su filosofía.

Los esquemas actuales presentan varios problemas en materia de seguridad y privacidad. También las empresas que los utilizan son dueños de nuestra identidad; resulta evidente que es necesario un cambio radical.

SSI vino a brindarle al propietario la gestión de sus datos e identidad, permitiéndole decidir que comparte, con quienes y por cuanto tiempo. Este paradigma facilita la transparencia. También ofrece una mejor adecuación a legislaciones vigentes como GDPR.

En la prueba de concepto se demostró que su aplicación puede ser viable; no obstante, es un ejemplo didáctico con varias salvedades y a una escala reducida.

Al momento de escribir este trabajo, el autor no ha encontrado casos de uso o soluciones SSI finalizadas; no obstante, la comunidad y varias organizaciones están infundiendo y trabajando arduamente para llevar adelante de manera exitosa esta filosofía.

En el capítulo tres se describieron desafíos de características técnicas, otras relacionadas al compromiso y a lo económico.

En cuanto a lo técnico, existe una arquitectura que aún no está madura, con problemas como la gestión de las claves, el acceso sin conexión y la facilidad de uso.

El proyecto Hyperledger Aries surgió en el 2019 con intenciones de solventar la interoperabilidad entre agentes y los distintos métodos DID; aunque es muy pronto para hablar de avances concretos cuando se sigue debatiendo que funcionalidades permanecen y cuáles van a Indy.

Los desafíos relacionados a la falta de compromiso y apoyo financiero se podrían superar si existiesen proyectos SSI finalizados exitosamente. Quizás en un futuro, la correcta implementación de DIDI o soluciones como las que busca aplicar el gobierno de Canadá [29] despierten el interés y el sustento económico de grandes empresas para este modelo en constante crecimiento.

A fin de cuentas, SSI solo tiene 4 años de existencia; por lo cual es normal que se presenten los desafíos planteados, probablemente surjan muchos más; no obstante, conforme se desarrolle, comprendan las ventajas y expanda su uso, es muy factible que este modelo sea el estándar elegido por todos.

Bibliografía

K. Cameron, «Kim Cameron's Identity Weblog,» 05 2005. [En línea]. Available: https://www.identityblog.com/?p=352. [Último acceso: 19 02 2020].

[1]

A. Preukschat y R. Drummond, Self-Sovereign Identity Descentralized Digital, Decentralized Digital Identity and Verifiable Credentials, Manning, 2020.

[2]

A. Preukschat y L. Carmona, *Identidad Digital Descentalizada,* España, 2019.

[3]

W. E. Forum, «A Blueprint for Digital Identity,» 2016.

[4]

Infobae, «Infobae,» 20 03 2018. [En línea]. Available: https://www.infobae.com/america/tecno/2018/03/20/7-datos-para-entender-el-escandalo-de-facebook-y-cambridge-analytica/. [Último acceso: 19 02 2020].

[5]

Verifiable Credentials Working Group, «W3C Standars,» [En línea]. Available: https://www.w3.org/2017/vc/WG/. [Último acceso: 20 02 2020].

[6]

C. Allen, «Life With Alacrity,» 25 04 2016. [En línea]. Available: http://www.lifewithalacrity.com/2016/04/the-path-to-self-soverereign-identity.html. [Último acceso: 29 02 2020].

[7]

S. Casas, «Universidad Politécnica de Madrid,» Youtube, 23 04 2019. [En línea]. Available: https://www.youtube.com/watch?v=34U2go3IMTQ&list=PLmSpCXWKxmqFXEK3o9NyrV08MaWS4D0HY. [Último acceso: 20 02 2020].

[8]

D. Reed, «SSIMEETUP,» 28 06 2018. [En línea]. Available: https://ssimeetup.org/decentralized-key-management-dkms-essential-missing-piece-ssi-puzzle-drummond-reed-webinar-8/.

[9]

D. Reed, «Linkdn,» [En línea]. Available: https://www.linkedin.com/in/drummondreed.

[10]

W3C Working Group Charter, «W3C,» 2019. [En línea]. Available:
11] https://www.w3.org/2019/09/did-wg-charter.html.

D. Reed, «SIMeetup,» 08 05 2018. [En línea]. Available:
12] Decentralized Identifiers (DIDs): The Fundamental Building Block of Self-Sovereign Identity – Drummond Reed – Webinar 2.

W3C, «Especificación del método Sovrin DID,» [En línea].
13] Available: https://sovrin-foundation.github.io/sovrin/spec/did-method-spec-template.html.

M. Schäffner, *Analysis and Evaluation of Blockchain-based Self-*
14] *Sovereign Identity System,* 2019.

W. C. C. Group, «W3C,» 29 02 2020. [En línea]. Available:
15] https://w3c-ccg.github.io/did-method-registry/.

M. Sabadello, «Introduction to DID Auth for SSI – Markus
16] Sabadello – Webinar 10,» SIMeetup, [En línea]. Available: https://ssimeetup.org/introduction-did-auth-markus-sabadello-webinar-10/.

Universidad Nacional del Sudoeste, «Universidad Nacional del
17] Sudoeste,» [En línea]. Available: https://certificados.upso.edu.ar/.

Signatura, «Signatura,» [En línea]. Available:
18] https://blog.signatura.co/identidad-digital-descentraliza-y-la-propuesta-de-signatura-el-caso-del-pasaporte-financiero-7a2c118f6128. [Último acceso: 11 04 2020].

M. Berman, «SSIMETUP,» [En línea]. Available:
19] https://ssimeetup.org/didi-ssi-public-blockchains-inclusion-argentina-latam-milton-berman-webinar-28/. [Último acceso: 11 04 2020].

A. D. Matteo, «Diario Bitcoin,» 16 08 2020. [En línea]. Available:
20] https://www.diariobitcoin.com/paises/sur-america/argentina/proyecto-didi-solucion-de-identidad-digital-autosoberana-argentina/. [Último acceso: 25 09 2020].

IProUP, [En línea]. Available:
21] https://www.iproup.com/innovacion/11915-didi-asi-es-el-programa-de-inclusion-digital-para-emprendedores-de-barrios-vulnerables. [Último

acceso: 11 04 2020].

22] T. L. Fundation, «Introduction to Hyperledger Blockchain Technologies,» Edx, [En línea]. Available: https://www.edx.org/es/course/introduction-to-hyperledger-blockchain-technologie.

23] Docker Inc, «Play with Docker,» [En línea]. Available: https://labs.play-with-docker.com/#.

24] VON, «IIWBook,» [En línea]. Available: https://vonx.io/how_to/iiwbook.

25] V. Baya, «CITI VENTURES,» 29 10 2019. [En línea]. Available: https://www.citi.com/ventures/perspectives/opinion/digital-identity.html.

26] L. C. P. Alex Preukschat, «Decentralized Digital Identity in the Spanish and Portuguese speaking world – Alex Preukschat – Webinar 50,» SSIMEETUP, [En línea]. Available: https://ssimeetup.org/decentralized-digital-identity-spanish-portuguese-speaking-world-alex-preukschat-webinar-50/. [Último acceso: 14 04 2020].

27] D. Hardman, «SSIMEETUP,» [En línea]. Available: https://ssimeetup.org/identity-quest-self-sovereign-identity-daniel-hardman-webinar-6/. [Último acceso: 15 04 2020].

28] S. J. J. S. Uwe Der, «Self-sovereign Identity – Opportunities and Challenges».

29] T. Bouma, «Medium,» [En línea]. Available: https://medium.com/@trbouma/canada-enabling-self-sovereign-identity-efcfda2aa044.

30] W3C Credentials Comunity, «W3C Working Draft,» 25 02 2020. [En línea]. Available: https://w3c.github.io/did-core/#did-documents. [Último acceso: 29 02 2020].

31] A. E. y. o. Markus Sabadello, «Introduction to DID Auth,» 26 97 2018. [En línea]. Available: https://github.com/WebOfTrustInfo/rwot6-santabarbara/blob/master/final-documents/did-auth.pdf.