Universitat Oberta de Catalunya

uoc.edu

Gestión soberana de identidades descentralizadas con Blockchain

Bernat Fernández Gonzalvo
Máster Interuniversitario de Seguridad en las TIC (MISTIC)
Sistemas de autenticación y autorización

Enric Hernández Jiménez
Víctor García Font

Enero 2019

Licencias alternativas (elegir alguna de las siguientes y sustituir la de la página anterior)

A) Creative Commons:

Esta obra está sujeta a una licencia de Reconocimiento-NoComercial-SinObraDerivada 3.0 España de Creative Commons

Esta obra está sujeta a una licencia de Reconocimiento-NoComercial-CompartirIgual 3.0 España de Creative Commons

Esta obra está sujeta a una licencia de Reconocimiento-NoComercial 3.0 España de Creative Commons

Esta obra está sujeta a una licencia de Reconocimiento-SinObraDerivada 3.0 España de Creative Commons

Esta obra está sujeta a una licencia de Reconocimiento-CompartirIgual 3.0 España de Creative Commons

Esta obra está sujeta a una licencia de Reconocimiento 3.0 España de Creative Commons

B) GNU Free Documentation License (GNU FDL)

C) Copyright

FICHA DEL TRABAJO FINAL

Título del trabajo:	*Gestión soberana de identidades descentralizadas con Blockchain*
Nombre del autor:	*Bernat Fernández Gonzalvo*
Nombre del consultor/a:	*Enric Hernández Jiménez*
Nombre del PRA:	*Víctor García Font*
Fecha de entrega (mm/aaaa):	01/2019
Titulación:	*Máster Interuniv. de Seguridad en las TIC*
Área del Trabajo Final:	*Sistemas de autenticación y autorización*
Idioma del trabajo:	*Castellano*
Palabras clave	*Self-Sovereign Identity, Sovrin, Hyperledger Indy*

Resumen del Trabajo (máximo 250 palabras): *Con la finalidad, contexto de aplicación, metodología, resultados i conclusiones del trabajo.*

La finalidad del Trabajo es ofrecer al lector una comprensión holística del paradigma de gestión soberana de la identidad. Para ello se analizan el resto de los paradigmas que actualmente existen y se presentan las ventajas del nuevo. Se explican también los detalles técnicos a un nivel suficiente para comprender el porqué actualmente la implantación de este paradigma es viable y para poder analizar técnicamente sus principios de funcionamiento.

Después de leer el Trabajo el lector debería ser capaz de profundizar en la tecnología utilizada hasta un nivel suficiente como para desarrollar o implementar soluciones vinculadas al ecosistema Sovrin, que es actualmente la plataforma de facto para la gestión soberana de la identidad.

El Trabajo analiza primero el dominio de la gestión soberana desde un punto de vista completamente funcional y teórico, para posteriormente trazar sus conceptos principales a la solución técnica y al entorno práctico.

El resultado del Trabajo es un documento autocontenido que, sin necesidad de consultar fuentes dispares, aporta un nivel de conocimientos significativo incluso para aquellas personas sin demasiado bagaje técnico y sin experiencia en el asunto.

Como conclusiones remarcables destacan la confirmación de la potencial viabilidad de las soluciones técnicas actuales, teniendo en cuenta la falta de madurez del ecosistema. Se demuestra que, a pesar de que técnicamente se puede implementar el paradigma de gestión soberana de la identidad, queda mucho trabajo por hacer, tanto a nivel pedagógico (concienciación de los usuarios) como a nivel técnico (desarrollo de herramientas).

Abstract (in English, 250 words or less):

The aim of the Project is to offer the reader a holistic understanding of the paradigm of self-sovereign identity management. To do this, the rest of the paradigms that currently exist are analysed and the advantages of the new one are presented. The technical details are also explained at a sufficient level to understand why the implementation of this paradigm is currently viable. The details explained also allows the reader to be able to analyse its operational principles technically.

After reading the Document, the reader should be able to delve into the technology used to an enough level to develop or implement solutions linked to the Sovrin ecosystem, which is currently the de facto platform for the self-sovereign identity management.

The Work first analyses the domain of self-sovereign identity management from a fully functional and theoretical point of view, and then traces its main concepts to the technical solution and its practical environment.

The result of the Project is a self-contained document that, without the need to consult different sources, brings a significant level of knowledge even for those people without too much technical background and without experience in the field.

Remarkable conclusions include the confirmation of the potential viability of current technical solutions, considering the lack of maturity of the ecosystem. It shows that, although the paradigm of self-sovereign identity management can technically be implemented, much work remains to be done, both at the pedagogical level (user awareness) and at the technical level (tool development).

ÍNDICE

LISTA DE FIGURAS

1. INTRODUCCIÓN

1.1 CONTEXTO Y JUSTIFICACIÓN DEL TRABAJO

El presente trabajo ahonda en la tendencia que están tomando los **Sistemas Gestión de la Identidad**. Dichos sistemas tratan de implementar en el mundo digital mecanismos análogos a los que en el mundo físico nos permiten demostrar que somos quién decimos ser.

Cuando hablamos de quién somos, realmente nos referimos a cómo somos, siendo este cómo lo que conforma nuestra identidad. Lo que nos identifica no es más que un **conjunto de características** asociadas a nuestro ser. Estas características son entendidas como atributos, de los que podemos distinguir distintos tipos (World Economic Forum, 2016).

Existen unos **atributos intrínsecos** a nuestra persona e inmutables en el tiempo. Estos pueden ser nuestra fecha o lugar de nacimiento, el nombre de nuestros progenitores, nuestra huella dactilar o nuestro código genético.

Existen **atributos acumulativos**, es decir que vamos acumulando con el paso del tiempo, como pueden ser nuestro estado civil, el número de hijos a nuestro cargo, nuestro historial médico, nuestro *curriculum* laboral o nuestro historial académico.

Por último, existen **atributos que nos asignamos** en un momento determinado, y que pueden verse o no modificados con el paso del tiempo. Estos podrían ser nuestro número del carné de identidad, nuestro lugar de residencia, nuestro número de teléfono, el cargo que ostentamos en la empresa en la que trabajamos o nuestro seguro de salud contratado.

El conjunto de atributos que tenemos vinculados, y el valor de cada uno de ellos en cada momento, es lo que **conforma nuestra identidad**. Tanto en el mundo físico como en el mundo digital, es importante el poder demostrar de manera fehaciente nuestra identidad, o al menos, determinados atributos de esta. Así, por ejemplo, si queremos poder ejercer nuestro derecho a voto, es necesario que demostremos que somos mayores de edad.

Para otorgar esa característica de veracidad a nuestros atributos se utilizan las **credenciales**, que no son más que documentos que **acreditan una aseveración**, en este caso el valor de un determinado atributo. En el mundo físico estas credenciales suelen ser documentos tangibles como, por ejemplo, nuestro carné de identidad expedido por el Estado del que somos nacionales.

En el **mundo digital** también son necesarias las credenciales, pero en este caso dejan de ser documentos físicos y pasan a convertirse en documentos “lógicos”. Es ahí donde empieza el reto para los sistemas de gestión de la identidad y donde se produce el **cambio de paradigma**.

El hecho de digitalizar nuestras credenciales puede tener varios fines, que van desde el simple hecho de disponer de una copia informatizada hasta el permitirnos realizar cualquier tipo de transacción de manera completamente telemática y segura para todas las partes.

En el primer caso nos bastaría con escanear nuestro documento de identidad, pero el valor real de nuestra credencial digital seria relativamente pobre, ya que la información contenida en la misma únicamente seria interpretable por otro ser humano, y la confirmación de su validez quedaría a su criterio (igual que con una credencial física).

Si queremos operar de manera telemática es necesario que las credenciales digitales puedan ser procesadas por máquinas, sin que se requiera de la intervención humana. Y si queremos operar de manera segura estas deben poder ser verificadas del mismo modo, sin necesidad del criterio y del juicio humano.

Con todo ello, para poder disponer de una **identidad digital** y para poder gestionarla, se hace necesario un ecosistema técnico que nos permita acumular nuestro catálogo de atributos, pudiendo además **demostrar su valor** de manera fehaciente a quien nos lo requiera. Con este fin han aparecido y han empezado a evolucionar los Sistemas de Gestión de la Identidad o IDMS (*IDentity Management Systems*).

Antes de concluir este preámbulo introductorio se hace necesario realizar algunos comentarios al respecto de la identidad digital.

El primero de ellos es que la identidad no es aplicable solo a personas, sino que lo es a **cualquier tipo de entidad**, sean individuos, organizaciones o incluso bienes. Cuando queremos realizar una transacción de manera telemática y segura puede ser necesario que nosotros expongamos cual es nuestra identidad, pero es igual de necesario que lo haga nuestra contraparte, y esta, no en pocas ocasiones, será o una empresa privada o una organización gubernamental.

El segundo de los apuntes está relacionado con dos conceptos que normalmente siempre van ligados a la gestión de la identidad. El primero de ellos es la **autenticación** y el segundo la **privacidad**. Es bueno no confundir estos términos y tener claro a qué hace referencia cada uno de ellos.

La autenticación, cuando se aplica a una persona, es el procedimiento por el cual se demuestra que esta es quién dice ser. Cuando se aplica a un documento es el proceso por el cual **se verifica que este es auténtico**. Por tanto, podemos autenticar a una persona verificando uno o varios de los atributos que conforman su identidad, y si este atributo depende de una tercera entidad, será necesario que esta previamente haya autenticado dicho atributo.

Así, si queremos autenticar a una persona mediante su carné de identidad digital, será necesario que el emisor de este certifique que dicho carné es auténtico. Para ello la persona deberá poseer un carné de identidad digital **certificado**.

En cuanto a la privacidad, en el contexto que nos ocupa, entendemos como tal a la característica que se les puede asignar a los atributos que conforman nuestra identidad para **regular quién tiene acceso** a los mismos. Así los atributos con un mayor nivel de privacidad serán únicamente accesibles por las entidades en las que nosotros confiemos de manera explícita. En cambio, los atributos con menor nivel

de privacidad serán accesibles de manera pública por cualquier entidad que pudiera estar interesada.

La gestión de la privacidad es la capacidad que tenemos como individuos de decidir qué información relativa a nuestra identidad compartimos, con quién la compartimos, con qué finalidad y en qué condiciones lo hacemos.

El Trabajo de Fin de Máster (TFM) que se desarrolla en el presente documento pretende analizar el estado del arte de los Sistemas de Gestión de la Identidad e implementar uno basado en los últimos avances tecnológicos, justificando así su viabilidad técnica y su utilidad en la vida cotidiana.

Para ello se expondrán brevemente los principales conceptos asociados a los IDMS, se analizarán las diferentes tipologías existentes a fecha de hoy y se detallará el proceso de implantación de uno de ellos simulando un entorno real.

Todo ello pretende ser una guía de conocimiento que permita al lector profundizar en el nivel de conocimiento acerca de los IDMS y concienciarlo de la importancia de la **buena gestión de la identidad digital**. Muchas veces es la falta de evolución de las herramientas lo que impide mejorar las prácticas actuales, y la gestión de la identidad es un claro ejemplo de ello.

La aparición de tecnologías disruptivas como el *blockchain* han permitido plantearse nuevos paradigmas técnicos que nos permiten dar un paso de gigante en las buenas prácticas en cuanto a la gestión de nuestra identidad digital, y que incluso, permiten esbozar nuevos paradigmas más justos para todos.

Difundir este conocimiento y hacer accesibles estas nuevas tecnologías es clave para romper la autocracia que actualmente imponen las principales corporaciones privadas del mundo digital en lo referente a la gestión de nuestra información personal.

El conocimiento también permite elevar el nivel de exigencia impuesto a nuestros gobiernos y a su elenco de entidades públicas que actualmente son el otro gran actor en lo referente al acceso a nuestros datos personales.

En la era de la información digital, y en un mundo que cada vez encuentra nuevas maneras de comerciar con ella, es objetivo del TFM el aportar un granito de arena tanto en el conocimiento como en la concienciación en lo relativo a nuestra identidad digital y a cómo gestionarla. Para ello que mejor que una implementación práctica que demuestre que ya tenemos a nuestro alcance herramientas acorde a un nuevo paradigma más justo que el actual.

1.2 OBJETIVOS DEL TRABAJO

Los principales objetivos del Trabajo pasan por:

1. Realizar un análisis de la evolución y del estado del arte actual de los Sistemas de Gestión de la Identidad (IDMS).
2. Asociar y describir los paradigmas de gestión de la información para cada una de las tipologías de IDMS.
3. Identificar las palancas tecnológicas que han permitido un avance significativo en dichos paradigmas describiendo sus características técnicas.
4. Definir un entorno de aplicación real para un IDMS basado en los últimos avances técnicos descritos, identificando sus componentes principales y detallando su arquitectura funcional.
5. Detallar la arquitectura técnica necesaria para implementar el IDMS en el entorno definido y describir a nivel técnico cada uno de sus elementos.
6. Implementar el IDMS sobre un entorno operativo garantizando que este es completamente funcional y que refleja las virtudes del nuevo paradigma de gestión de la identidad.
7. Justificar la extrapolación de la prueba de concepto a un entorno real, real tanto a nivel funcional como a nivel técnico.
8. Sumarizar las conclusiones extraídas y plantear los próximos pasos necesarios para seguir fomentando el cambio de modelo en cuanto a la gestión de la identidad digital.

1.3 ENFOQUE Y MÉTODO SEGUIDO

El enfoque del Trabajo pasa por realizar una primera labor de investigación para asentar los conceptos y conocer el estado del arte en la materia de la gestión de la identidad digital, y posteriormente por aplicar dichos conocimientos en el desarrollo de un caso práctico, de manera que pueda demostrarse la usabilidad de las nuevas tecnologías y la aplicabilidad de los conceptos teóricos analizados.

Así, la estrategia del Trabajo contempla un primer hito teórico orientado a divulgar el conocimiento existente relativo al nuevo paradigma de gestión de la identidad digital y a reconocer las capacidades disruptivas que ofrecen las nuevas tecnologías vinculadas a las cadenas de bloques (*blockchain*), que permiten alcanzar dicho paradigma.

Para ello se consultará la bibliografía disponible, tratando de digerir y sintetizar toda la información contenida, poniendo a disposición del lector un resumen bien documentado que le permita formarse un criterio al respecto de la materia que nos ocupa.

Seguidamente se tratará de demostrar la viabilidad en lo referente a la implementación de estas nuevas tecnologías mediante la utilización práctica de las últimas herramientas aparecidas. Además de conseguir

una implementación completamente funcional, se centrarán los esfuerzos en asegurar la trazabilidad entre los conceptos teóricos y los prácticos.

Finalmente se expondrán las conclusiones extraídas, centrándose estas en evaluar el estado de madurez de las soluciones prácticas implementadas. Para ello se tendrán en cuenta las barreras técnicas a superar y los conocimientos necesarios para aplicarlas en un entorno real, a pequeña, mediana y gran escala. Es decir, la viabilidad de utilizar dichas herramientas en el día a día de pequeñas y medianas empresas (PYME), grandes corporaciones y entidades gubernamentales.

1.4 PLANIFICACIÓN DEL TRABAJO

El Trabajo se estructura en cuatro entregas incrementales asociadas a cuatro PEC. Se describe a continuación el contenido de cada una de ellas:

Entregable	Fecha	Contenido
PEC1	08/10/2018	Entrega con la planificación inicial del Proyecto, donde se esbozarán los objetivos y la estructura del mismo. Se detallará el enfoque y el método a aplicar y se planificarán las entregas. Servirá para orientar el resto del Proyecto y para confirmar la dirección a tomar y el alcance de este.
PEC2	05/11/2018	Entrega que recogerá y expondrá el análisis teórico de los IDMS y de los distintos paradigmas de gestión de la identidad. Identificará los componentes principales a considerar en la implementación práctica y describirá las principales herramientas a utilizar, así como las tecnologías vinculadas. Servirá para asentar conocimientos y para definir el escenario funcional a implantar en la parte práctica del trabajo.

Entregable	Fecha	Contenido
PEC3	03/12/2018	Entrega que culminará con el diseño técnico de la solución y con la puesta a punto de la plataforma tecnológica necesaria para la posterior implementación de dicha solución. Servirá para preparar el escenario técnico y para poner en marcha los principales componentes de la parte práctica del trabajo.
PEC4	31/12/2018	Entrega que contemplará la puesta en marcha del escenario técnico y la validación de todas sus funcionalidades. Arrojará también las conclusiones extraídas del Trabajo y establecerá los posibles próximos pasos a seguir en caso de querer avanzar en el camino trazado. Servirá para concluir y formalizar la entrega del Trabajo, aglutinándose en dicha entrega todo el contenido generado.

Tabla 1. Planificación de los entregables del Trabajo

1.5 BREVE SUMARIO DE PRODUCTOS OBTENIDOS

Los productos obtenidos durante la elaboración del trabajo básicamente serán la propia memoria y el conjunto de componentes técnicos necesarios para implementar el caso práctico. El detalle de estos componentes se abordará en los próximos apartados.

1.6 BREVE DESCRIPCIÓN DE LOS OTROS CAPÍTULOS DE LA MEMORIA

Los capítulos que conformarán la memoria del Trabajo de Fin de Máster serán los siguientes (susceptibles de ser modificados durante la elaboración del mismo):

- Capítulo 1. Introducción: desarrollará la introducción al Trabajo, exponiendo su contexto, sus objetivos, su enfoque y su planificación.

- Capítulo 2. Análisis de la situación actual: Análisis de la situación actual: donde se describirá desde un punto de vista teórico la filosofía de los sistemas de gestión de la identidad y se analizarán las tendencias y el estado del arte de estos.

- Capítulo 3. Descripción de la solución técnica: Sovrin - Identity for All: donde se analizará el funcionamiento de las herramientas técnicas y de las tecnologías necesarias para implantar el nuevo paradigma de la gestión de la identidad (*Blockchain*, Sovrin, etc.).

- Capítulo 4. Diseño del Caso de Uso para el entorno Práctico: recogerá el diseño del ecosistema técnico que permitirá implantar la solución tecnológica asociada al nuevo paradigma de gestión de la identidad, haciéndose uso de las herramientas necesarias. Se identificarán cada uno de sus componentes, así como las funcionalidades principales a satisfacer.

- Capítulo 5. Análisis Técnico del Caso de Uso en el entorno Práctico: donde se describirá la implementación técnica realizada y se analizará su funcionamiento, asegurándose que se satisfacen todas las funcionalidades definidas y que se da respuesta al paradigma establecido.

- Capítulo 6. Conclusiones: que recogerá las conclusiones del Trabajo, tratando de confirmar la viabilidad de la solución práctica implantada y su adecuación a un entorno real a diferentes escalas.

[página intencionadamente en blanco]

2. ANÁLISIS DE LA SITUACIÓN ACTUAL

Se analizan a continuación los distintos **paradigmas de gestión de la identidad** que la técnica y la evolución han permitido implantar en la actualidad, desde sus inicios hasta las nuevas tendencias que prometen revolucionar la manera que hasta ahora tenemos de demostrar quienes somos.

2.1 LA NECESIDAD INICIAL Y LOS PARADIGMAS CONVECIONALES

Igual que la mayoría de las aplicaciones, los **sistemas de gestión de la identidad** (IDMS) han formado parte de la rápida evolución experimentada por el entorno informático desde mediados del siglo XX.

Estos nacieron a raíz de la implantación de entornos de cómputo colaborativos, en los que varias personas o varios grupos podían trabajar en un solo "súper-equipo", de manera que se compartían los costes, tanto de inversión como de operación de las grandes máquinas que entonces eran los centros de procesamiento. Para poder repercutir y repartir estos costes se hizo necesario el identificar quién había utilizado la infraestructura y durante cuánto tiempo lo había hecho. Para dicha identificación era suficiente con introducir un número identificativo.

2.1.1 LA GESTIÓN INTERNALIZADA DE LA IDENTIDAD

El siguiente paso se dio cuando los entornos de computación colaborativos empezaron a ofrecer más funcionalidades y aparecieron los primeros clientes remotos. Entonces se hizo necesario, además de identificar a los diferentes usuarios, darles distintos privilegios que permitieran regular los permisos a aplicar a la hora de acceder a las funcionalidades y a la hora de compartir información. Con los sistemas distribuidos, se requiere disponer de un listado centralizado de usuarios para cada entorno colaborativo o dominio. En base a los atributos de cada usuario, y después de un proceso de autenticación normalmente basado en la combinación de un identificador y una palabra clave, se concede acceso a unos servicios determinados (se da autorización). En este ecosistema el servidor principal hace las veces de *Identity Provider* (IdP) y de *Relaying Party* (RP). La función de IdP es la que está a cargo de almacenar los atributos asociados a cada usuario y que sirven para identificarlo. La función de RP es la que proporciona acceso a una aplicación segura, es decir, a una aplicación que no admite un uso anónimo o no identificado (el *Relaying Party* necesita confiar en las credenciales identificativas para dar acceso). El hecho de centralizar ambas funciones en un punto común interno (el servidor central) para cada dominio es el que caracteriza al **paradigma de gestión internalizada de la identidad** (World Economic Forum, 2016).

Este paradigma sigue vigente hoy en día incluso en las soluciones más contemporáneas, como pueden ser los servicios de nube privados o de SaaS (*Software as a Service*). En estos sistemas los clientes pueden interaccionar desde cualquier lugar después de identificarse de manera expresa. Es en los servidores de dichos servicios donde se realiza la

autenticación de los usuarios, se recuperan sus atributos y en base a los mismos se garantiza un acceso seguro con unos privilegios u otros (se da autorización).

2.1.2 LA GESTIÓN CON AUTENTICACIÓN EXTERNALIZADA

El siguiente paso evolutivo consistió en separar estas dos funcionalidades manteniéndose la aplicación y sus mecanismos de acceso seguro centralizados (RP) pero permitiéndose la autenticación y el almacenamiento de los atributos identificativos de los usuarios (IdP) en otros puntos. De esta manera no es necesario que el proveedor de la aplicación disponga de los atributos de los usuarios, y además se permite que estos sean gestionados por diferentes entidades, por ejemplo, según su naturaleza. Así se consigue una segmentación en la gestión de atributos, favoreciéndose la especialización y limitándose las posibilidades de acceso cruzado a los mismos (agregación). Esta configuración es la que caracteriza al **paradigma de gestión de la identidad con autenticación externalizada** (World Economic Forum, 2016).

Ejemplo de la aplicación de este paradigma pueden ser los servicios públicos accedidos de manera digital (función de RP), que requieren de la autenticación fehaciente de los ciudadanos. Para ello puede utilizarse la solución del certificado digital (caso de España y la Agencia Tributaria) o puede permitirse que un tercero de confianza identifique y autentifique (función de IdP) a cada usuario. En el caso británico (GOV.UK) son entidades como los bancos o las empresas de telecomunicaciones las que utilizan los atributos y las cuentas de sus clientes para identificarlos y autenticarlos. Estas son entidades de confianza porqué se asume que para dar de alta a sus clientes y aceptarlos en sus dominios, ya han realizado un conjunto de acciones de verificación que garantizan un nivel de confianza suficiente. Este conjunto de acciones se engloba en un proceso denominado *Know Your Customer* (KYC), que suele requerir de la presentación de documentación identificativa oficial y de referencias laborales, de ingresos, etc.

2.1.3 LA GESTIÓN CENTRALIZADA DE LA IDENTIDAD

Otra estrategia para gestionar la identidad y el acceso seguro a servicios digitales pasa por centralizar el rol de IdP y distribuir la función de autenticación y el rol de RP. En este caso todos los atributos de los usuarios están centralizados en una entidad de confianza (función IdP), que suele ser un organismo público. Son los proveedores de servicios, públicos o privados, los que interaccionan con el proveedor de identidad para obtener los atributos necesarios que les permitan gestionar el proceso de autenticación y garantizar un acceso seguro (función RP) a dichos servicios. De esta manera el usuario sabe que tiene todos sus atributos (datos personales) centralizados en un único punto, y se le facilita su administración, ya que para que un tercero tenga acceso a los mismos puede requerirse su consentimiento. Una vez un conjunto de atributos se transfiere a un RP este los utiliza tanto para gestionar su propio proceso de autenticación (por ejemplo, utilizando una dirección de

correo electrónico o un número de identificación nacional) y para fines operativos o comerciales (por ejemplo, utilizando el código postal de residencia o el año de nacimiento). Este planteamiento es el que fundamenta el **paradigma de la gestión centralizada de la identidad** (World Economic Forum, 2016).

En los Países Bajos existen diferentes proveedores públicos de servicios digitales y todos ellos obtienen los atributos de los usuarios del Registro Civil (de ámbito nacional pero gestionado a nivel municipal). Cuando los ciudadanos se inscriben y aportan información sobre sus atributos (datos personales) pueden solicitar su identificador digital DigiD (https://www.digid.nl), que pasa a ser un atributo más de su identidad digital. Para obtener el DigiD es necesario que los ciudadanos se autentiquen como nacionales de los Países Bajos mediante su pasaporte o su carné de identidad. Este atributo (DigiD) es posteriormente utilizado por los proveedores públicos de servicios digitales para autenticar a los usuarios (por ejemplo, con el DigiD y una contraseña) y garantizarles un acceso seguro a sus aplicaciones.

2.1.4 LA GESTIÓN CON AUTENTICACIÓN FEDERADA

Con el fin de mejorar la experiencia del usuario, y evitarle el tener que autenticarse frente a cada RP (aun y estando sus atributos centralizados en un único IdP), los sistemas de gestión de la identidad han dado otro paso evolutivo. En este caso, los proveedores de identidad (IdP) se asocian o federan aglutinados bajo un dominio común. La federación contempla a un IdP con el rol proveedor primario, que es el que almacena los atributos de los usuarios. No obstante, el resto de IdP federados tienen la capacidad de proveer a los RP los atributos necesarios para autenticar a los usuarios. De este modo los proveedores de servicios (RP) pueden elegir a cualquiera de los IdP federados para autenticar a sus usuarios, evitándose el tener que ligarse a uno en concreto. Si el servicio requiere de atributos adicionales estos son suministrados por el IdP primario, que ya es conocedor de la autenticación positiva del usuario, puesto que está federado en el mismo domino que el IdP que ha llevado a cabo el proceso de autenticación. Así, una vez el usuario se ha autenticado en una aplicación (RP), el IdP primario puede proporcionar los atributos necesarios a cualquier otra aplicación que forme parte del dominio federado, dándose al usuario por autenticado sin necesidad de volver a solicitarle sus credenciales. Este es el **paradigma de gestión de la identidad basado en la autenticación federada** (World Economic Forum, 2016).

Este paradigma está actualmente muy extendido, ya que es la base del procedimiento de *Single Sign On*, un mecanismo de autenticación con gran implantación en entornos corporativos. Este procedimiento permite que el usuario de un sistema operativo (por ejemplo, Microsoft Windows®) deba ingresar una sola vez sus credenciales para tener acceso a todas las herramientas corporativas. En este caso el sistema operativo actúa como IdP primario, almacenando los atributos del usuario, normalmente a través de un servicio de directorio (*Active Directory*) accedido por LDAP. Una vez el usuario se ha autenticado en el sistema operativo (que en este

caso actúa también de RP), el resto de las aplicaciones federadas (el resto de RP) utilizarán esta autenticación positiva como válida para dar acceso a sus servicios. Para ello el sistema operativo (en su rol de IdP primario) les notificará que ha existido esta autenticación positiva y les facilitará los atributos que necesiten obteniéndolos del directorio. Estas aplicaciones federadas podrán ser tanto aplicaciones locales (por ejemplo, acceso a Microsoft Word® o a Microsoft Project®) como aplicaciones remotas centralizadas (por ejemplo, acceso a SAP® o a SalesForce®) o como aplicaciones web (por ejemplo, acceso al servidor web de correo electrónico).

2.1.5 LA GESTIÓN DESCENTRALIZADA DE LA IDENTIDAD

Si combinamos de múltiples proveedores de identidad (IdP) y de varios proveedores de servicios con acceso seguro (RP), se crea un modelo en el que distintas entidades pueden servir atributos y/o autenticar a diferentes proveedores, democratizando el mercado entre unos y otros. Con este modelo no es necesario que un RP se suscriba a una federación de IdP concreta, sino que puede elegir al proveedor de identidad que más le convenga. También puede elegir a un proveedor para resolver la autenticación y a otro distinto para obtener los atributos requeridos. Así, por ejemplo, un proveedor de servicios digitales (RP) no deberá “obligar” a su cliente a proporcionar sus datos personales a una federación concreta, o no se verá forzado a obtenerlos de un IdP único.

El cliente podrá elegir en qué entidad confía para proporcionar sus datos y el proveedor de servicios podrá integrase con esta entidad sin mayor problema. Esto da una gran flexibilidad tanto a los usuarios como a los proveedores de identidad (IdP) y a los proveedores de servicios (RP). Lo único que debe existir es un estándar común que todas las partes deban aplicar para implementar los mecanismos de recuperación de atributos y de autenticación (protocolos abiertos de este tipo pueden ser OpenID o Oauth). Esta configuración más abierta da lugar al **paradigma de gestión descentralizada de la identidad** (World Economic Forum, 2016).

Ejemplo de la implementación de este paradigma son las redes de telefonía móvil, que básicamente ofrecen servicios de comunicación por voz o servicios de transmisión de datos. En este caso los operadores de telecomunicaciones que disponen de cobertura en un lugar determinado actúan como RP. Los operadores de telecomunicaciones propietarios de la SIM instalada en los terminales móviles actúan como IdP. Es la propia SIM la que almacena los atributos del cliente (por ejemplo, el número de teléfono), y la que, mediante un protocolo estandarizado los suministra al RP una vez finalizado el proceso de autenticación. Como es bien sabido, tenemos multitud de operadores de telecomunicaciones que ofrecen servicios (cobertura) en distintos lugares (actuando como RP) y multitud de operadores ofreciendo SIM para terminales (actuando como IdP). Cualquier combinación de operadores (por ejemplo, red Telefónica® con SIM Vodafone®, o SIM Orange® con red Deutsche Telekom®) permite mantener el servicio al cliente gracias a los protocolos de autenticación e intercambio de atributos estandarizados, dándole al cliente la posibilidad

de elegir. En este caso es la asociación GSMA® la encargada de redactar los estándares técnicos que garantizan la interoperabilidad y que permiten la gestión descentralizada de las identidades.

Como se ha visto, los diferentes paradigmas de gestión de la identidad satisfacen varias necesidades o introducen determinadas mejoras dependiendo de su ámbito de aplicación. No obstante, todos ellos comparten algunas **características comunes**, hasta la fecha difíciles de cambiar.

- Los atributos de los usuarios (datos personales) siempre están **en manos de** (al menos) **un tercero**.

- Los mecanismos para proveer los atributos fehacientes siempre pasan por un **procedimiento no digitalizado** que requiere de la presentación de documentación física y de su evaluación "analógica", lo que siempre deja cabida al fraude.

- La mayoría de las veces los terceros tienen **acceso a más atributos** (datos personales) de los estrictamente necesarios para prestar sus servicios.

- Resulta relativamente sencillo vincular los atributos de los que disponen los terceros y **agregar datos personales** que revelen más información de la autorizada inicialmente.

- Es prácticamente imposible **controlar de manera fiable** el intercambio de información que se produce entre terceros y el uso que estos hacen de dicha información.

- Los atributos (datos personales) acaban proporcionando mucho **más valor a los terceros** que los explotan que a los propios usuarios que los ceden.

Todas estas características tienen un potencial impacto negativo para los usuarios propietarios de la identidad y para los proveedores, tanto de identidad como de servicios. Dicho impacto potencial va asociado a los riesgos que para los usuarios supone el no disponer del control de sus atributos (datos personales) y que para los proveedores supone el no disponer de una información confiable de sus clientes. No obstante, hasta la fecha no ha sido posible implementar ningún nuevo paradigma que mitigue estos riesgos.

2.2 EL NUEVO PARADIGMA Y SUS APORTACIONES

Buscando un modelo de gestión de la identidad más justo para los usuarios se plantea un paradigma en el que el gestor de los atributos (datos personales) sea su propietario, y no un tercero. De esta manera cada uno puede ser dueño de su propia identidad digital, agregando todos sus datos y conformando una identidad integral, o manteniéndolos desagregados conformando tantas "micro-identidades" diferentes como sea necesario.

Con este enfoque es cada usuario quién va generando sus atributos digitales y quién decide con quién compartirlos, de manera que la gestión de la identidad pasa a ser completamente soberana para cada individuo (u organización).

2.2.1 LOS PRINCIPIOS DE LA GESTIÓN SOBERANA

Para comprender el enfoque de este nuevo paradigma es necesario volver a los conceptos radicales de la gestión de la identidad en su dominio físico.

Cuando pensamos en cómo gestionamos nuestra identidad en el mundo físico podemos ver que quién somos, de cara al mundo, queda determinado por un conjunto de evidencias debidamente documentadas y certificadas por terceras entidades. Cuando alguien quiere verificar nuestra identidad, realmente no está confiando en nosotros, sino que confía en esas terceras entidades que han certificado las evidencias que presentamos. Esto es así, por ejemplo, cuándo queremos demostrar nuestra nacionalidad, nuestro nivel de estudios, nuestro nivel de ingresos o nuestro estado de salud.

Cuando pretendemos acceder a un servicio público, como podría ser el de búsqueda de empleo, es condición imprescindible el ser nacionales del país de residencia o el demostrar la residencia si estamos en un país extranjero. Para ello debemos presentar nuestro documento nacional de identidad (DNI) o un documento donde se nos identifique como extranjero residente (NIE). En ambos casos se trata de evidencias que vienen certificadas (mediante un sello distintivo) por el Gobierno nacional. Evidentemente el prestatario del servicio público confía plenamente en el Gobierno, así que dará por buena la evidencia presentada (DNI o NIE) siempre que pueda confirmar su autenticidad.

Cuando presentamos la candidatura a un empleo, el empleador requerirá un nivel de estudios determinado que nosotros demostraremos mediante la presentación de unas evidencias certificadas por una entidad reconocida (por ejemplo, una universidad) en la que él deberá confiar para darlas por buenas.

Cuando queramos pedir financiación para comprar una vivienda, el banco nos solicitará alguna evidencia que demuestre nuestra estabilidad laboral y nuestro nivel de ingresos. Para ello nos pedirá una copia de nuestro contrato y de nuestras últimas nóminas. No obstante, puede que el banco no confíe de manera explícita en la empresa en la que trabajamos, y para minimizar riesgos nos pida también un informe de nuestra vida laboral emitida por una entidad gubernamental (la Agencia

Tributaria). En este caso el banco requiere de un nivel de confianza mayor que el proporcionado por las evidencias certificadas por nuestra empresa.

Finalmente, si queremos contratar un seguro médico, el proveedor del servicio nos solicitará un informe médico oficial previo a la contratación. Probablemente este informe estará certificado por un médico de un organismo público (Seguridad Social), pero aun y así, el seguro exigirá un nivel de confianza mayor para minimizar riesgos. En este caso incluirá un periodo de carencia el cual le permitirá verificar por si mismo que no tenemos problemas de salud previos, generándose una evidencia auto-certificada, por tanto, con el máximo nivel de confianza.

Como vemos la verificación de la identidad no es más una cadena de confianza entre entidades que intercambian evidencias certificadas. El disponer de estas evidencias certificadas es lo que nos permite conformar nuestra identidad, permitiéndonos demostrar a los demás quiénes somos. En el caso de la identidad digital, todas estas evidencias son evidencias nativamente digitales, es decir, que se generan desde un primer momento ya en formato digital (no son evidencias físicas digitalizadas). El poder gestionar de manera soberana nuestra propia identidad significa el poder elegir de qué evidencias disponemos y el poder decidir con quién las compartimos. Así es como empieza a conformarse el ecosistema para el **paradigma de la gestión soberana de la identidad**.

2.2.2 LOS ROLES EN EL ECOSISTEMA DEL NUEVO PARADIGMA

Como gestores soberanos de nuestra identidad, lo primero que necesitamos es disponer de las evidencias que consideremos oportuno. Como hemos visto, una evidencia no tiene valor como tal hasta que no está certificada por una entidad en la que se confíe. Así, lo que necesitaremos en conseguir esa certificación y para ello presentaremos una petición a la entidad certificadora. En este nuevo ecosistema, a esta petición se la identifica como ***Claim***, y a la entidad que deberá certificarla para darla por buena (convertirla en evidencia), se la identifica como ***Issuer***. Finalmente, a la entidad que nos ha requerido que demostremos nuestra identidad se la identifica como ***Verifier***, ya que será la que verifique la validez de nuestras evidencias. Nosotros que presentamos las evidencias, como propietarios y gestores de los *Claim,* quedaremos identificados como ***Owners*** (o *Holders*).

Remarcar que para verificar la validez de las evidencias el *Verifier* comprobará que efectivamente las evidencias han sido certificadas por el *Issuer*, pero además deberá tener una relación de confianza con este. Del mismo modo el *Issuer* solo certificará los *Claim* presentados si realmente puede cerciorarse de que las peticiones son válidas y lícitas, ya que de otro modo su reputación se vería perjudicada y podría llegar a perder las relaciones de confianza que tuviera con los *Verifiers*, por lo que ningún *Owner* querría utilizar sus servicios de certificación de *Claims*.

Siguiendo con la analogía presentada, en el caso de la contratación por parte de una empresa privada, los roles quedarían asignados de la siguiente manera:

- El empleador asumiría el rol de *Verifier*, quién solicita al candidato que presente sus credenciales relativas a su nivel de estudios.

- El candidato tendrá el rol de *Owner* y será quien solicitará a la universidad que emita una evidencia que demuestre su titulación.

- La evidencia de la titulación universitaria será el *Claim* que el *Owner* presentará al *Verifier* para demostrar su identidad.

- La universidad asumirá el rol de *Issuer* emitiendo la titulación universitaria certificada para que pueda ser utilizada como evidencia.

Cuando quisiéramos solicitar financiación para la compra de una vivienda, el banco tomaría el rol de *Verifier*, nuestro empleador tomaría el rol de *Issuer*, y en este caso el *Claim* sería nuestro contrato de trabajo o nuestra nómina. Nosotros como *Owners* siempre seremos los encargados de adquirir los *Claims* certificados, de almacenarlos y de presentarlos cuando sea oportuno.

Obviamente el definir y asignar los roles del nuevo ecosistema no es suficiente para que este soporte un paradigma que solucione los problemas que presentan los sistemas actuales. Para ello es necesario dotarnos de mecanismos con los que:

- Podamos relacionarnos de manera confidencial con *Issuers* y con *Verifiers.*

- Los *Issuers* puedan certificar los *Claims* de una manera verificable por parte de los *Verifiers.*

- Los mecanismos de certificación no requieran de una Autoridad de Certificación (CA) centralizada que nos haga perder la soberanía.

Para alcanzar estos objetivos el ecosistema del nuevo paradigma requiere de una serie de componentes técnicos clave que se exponen a continuación.

2.2.3 LOS COMPONENTES CLAVE DEL NUEVO ECOSISTEMA

Tratando de democratizar la gestión de la identidad se crea un ecosistema en el que existen roles tanto de *Issuers* como de *Verifiers*, dónde una misma entidad puede asumir uno u otro dependiendo del contexto, de la situación y del momento. Además, la relación de confianza entre ellos puede regularse con los principios del libre mercado, basándose en la competencia, la calidad de sus servicios, la reputación, y, en definitiva, la libre elección de los consumidores.

No obstante, el dinamismo de estos roles hace que sea imprescindible el poder relacionarnos con estas entidades de manera confidencial si queremos preservar nuestra privacidad, y por tanto ser

verdaderamente soberanos a la hora de gestionarla. Si nos identificamos con nombre y apellidos cuándo nos dirigimos al banco para solicitar una hipoteca y con el mismo nombre y apellidos nos dirigimos a la universidad para solicitar nuestro título universitario, será fácil para un tercero cruzar los datos y trazar nuestras interacciones.

Ni la universidad necesita saber que estoy hipotecado para emitir mi titulación, ni el banco necesita conocer mi nivel de estudios para otorgarme financiación (a priori le es suficiente con conocer mi condición laboral y mi nivel de ingresos). Pero ¿podemos relacionarnos con ellos de manera segura a través de una infraestructura de acceso público sin necesidad de publicar ninguno de nuestros datos personales?

Imaginemos que disponemos de un identificador único (UID), que es un número aparentemente aleatorio, y que lo utilizamos para identificarnos frente al banco. Tenemos otro número diferente para identificarnos frente a la universidad. Las interacciones para obtener la evidencia del título universitario irán asociadas al UID de la universidad, y las asociadas a la solicitud de la hipoteca al UID del banco, por lo que no habrá trazabilidad entre ellas. Pero ¿cómo lo hace entonces la universidad para saber quién somos y buscarnos en su histórico de alumnos? ¿Y cómo se asegura de que está entregando dicho certificado al alumno que decimos ser? Todo ello además sin tener que pasar por una entidad de certificación centralizada y garantizando la validez legal de las evidencias.

Los conceptos y componentes clave para poder implementar una solución técnica a estos problemas son:

- Los **Identificadores Descentralizados** (DID): se trata de identificadores únicos, con una prácticamente imposible probabilidad de colisión (no se repiten), que permiten a los participantes del ecosistema (*Issuers*, *Owners* y *Verifiers*) generarse y administrarse su identidad sin necesidad de que exista ninguna entidad centralizada.

- Las **credenciales verificables**: que mediante los mecanismos de firma digital permiten asegurar la integridad de las evidencias (que no han sido modificadas) y el no repudio de su emisión (permiten demostrar fehacientemente quién las ha emitido).

- La **gestión distribuida de claves**: que permite trabajar con una infraestructura de clave pública (PKI) si necesidad de una entidad de certificación que certifique la relación de los participantes con sus claves públicas.

La tecnología de firma digital y de infraestructura de clave pública es una tecnología ampliamente conocida y madura, pero lo que realmente aporta el cambio disruptivo que ha permitido el nacimiento de este nuevo paradigma es el atributo “descentralizado”. La aparición de la tecnología de cadena de bloques (o *blockchain*) ha sido la palanca que ha permitido plantearse el prescindir de las entidades que hasta ahora constataban la asociación entre identificadores (ID) y usuarios, y entre clave públicas y sus titulares.

La idea es bastante sencilla: tenemos una base de datos de acceso público en la que podemos garantizar que los registros que se generen nunca podrán ser modificados (garantizándose su integridad). Además, esta base de datos estará replicada (garantizándose su disponibilidad) y solo podrán escribir en ella aquellos que hayan demostrado actuar de buena fe y en pro de la comunidad (garantizándose la veracidad). Todos estos principios son los que están detrás de lo que hoy en día se conoce como ***Distributed Ledger Technology*** o DLT.

El primero de los requisitos (integridad) se alcanza mediante la utilización de la cadena de bloques, en la que el contenido de un bloque (podríamos hacer la analogía con el contenido de un registro en una base de datos) se vincula criptográficamente al contenido del siguiente bloque. De esta manera, para modificar el contenido de un bloque determinado deberíamos modificar criptográficamente el contenido de todos sus bloques sucesores, con el coste computacional que ello supone.

El segundo de los requisitos (disponibilidad) lo conseguimos mediante los mecanismos y protocolos de consenso, con la existencia de múltiples nodos encargados de introducir los nuevos bloques de información capaces de ponerse de acuerdo para coordinarse y distribuirse la información actualizada entre sí. Todos los nodos tratarán de actualizar los bloques, pero el primero que lo consiga lo notificará a los demás y estos dejarán de intentarlo y actualizarán su información. De este modo nunca se depende de un solo nodo, ni para actualizar la información ni para mantenerla disponible.

El tercer requisito (veracidad) se consigue regulando la posibilidad de establecerse como nodo generador de bloques, es decir, dando permisos para poder escribir en el *ledger*. A los *ledgers* de este tipo se les conoce como ***public permissioned ledger***, ya que el acceso a los mismos es público, pero no así la posibilidad de validar su información (escribir nuevos bloques). Esto significa que debe existir una organización que administre estos permisos, pero no tiene por qué ser necesariamente la que decida a quién se le conceden.

Si añadimos el *public permissioned ledger* al ecosistema técnico vinculado al paradigma de la gestión soberana de la identidad, ya tenemos todos los componentes necesarios para hacerlo técnicamente viable. Será el *ledger* el que permitirá prescindir de la centralización, ya que en él podremos almacenar todos los identificadores (DID) de los actores del sistema, y será donde podremos trazar la propiedad de las claves públicas y asociarlas con estos identificadores. No necesitaremos una *Certification Authority* (CA) que de fe de dicha relación.

2.2.4 LOS PRINCIPIOS DE FUNCIONAMIENTO DEL ECOSISTEMA

Con todos los componentes clave del ecosistema establecidos, los principios funcionales para poder implementar un sistema de gestión soberana de la identidad serán los siguientes:

A. La comunicación entre los actores se realizará de manera bilateral y punto a punto (peer-to-peer o P2P), y ambos interlocutores se identificarán con un identificador único descentralizado (DID). Cada actor decidirá si utiliza el mismo identificador para todas las relaciones o si utiliza uno distinto para cada una (esto sería lo recomendable para evitar la posibilidad de que un tercero pudiera trazar sus comunicaciones).

B. Cada identificador (DID) tendrá asociados un par de claves, una pública y una privada. El propio identificador y su clave pública estarán disponibles en el *ledger*, de manear que podrán ser consultados por el resto de los actores. La clave privada únicamente estará en poder del propietario del identificador.

C. Los actores intercambiarán evidencias que demostrarán la veracidad de los atributos que conforman cualquier identidad. Los *Owners* solicitarán las evidencias a los *Issuers* para presentarlas a los *Verifiers*. Estos últimos las aceptarán si pueden autenticar su emisión por parte del *Issuer* y si siempre que mantengan una relación de confianza con él.

D. Para establecer una conexión entre participantes deberá procederse con una autenticación mutua. El participante que inicie la comunicación facilitará su identificador (DID) a la contraparte. La contraparte obtendrá del *ledger* la clave pública vinculada a ese identificador y firmará su mensaje de respuesta con dicha clave, asegurándose por tanto de que únicamente quién posea la clave privada podrá procesarla.

E. Asegurada la conexión y la autenticidad de las partes, el *Owner* solicitará la evidencia al *Issuer*. Este generará la evidencia y la firmará con su clave privada. Cuando el *Owner* entregue la evidencia al *Verifier*, este solicitará la clave pública del *Issuer* al *ledger*. El *ledger* le entregará la clave pública al *Verifier* y este podrá comprobar que efectivamente dicha firma solo podría haber sido generada por el poseedor de la clave privada vinculada a esa clave pública. Con eso podrá garantizar que realmente el *Issuer* está certificando que los atributos pretendidos por el *Owner* son verídicos.

F. El *Owner* dispondrá de estas evidencias firmadas y podrá presentarlas a todos los *Verifiers* que las soliciten. Si un *Verifier* solicita una evidencia muy específica, el *Owner* podrá solicitarla al *Issuer*, en el formato en el que la necesite. Así, será el *Owner* quién decida qué información entrega al *Verifier*, pudiéndose ceñir a lo estrictamente necesario. Por ejemplo, para demostrar la mayoría de edad, no es

necesario revelar cuantos años se tienen, o cual es la fecha de nacimiento. Con esto aparece el concepto de ***Zero Knowledge Proof*** (ZKP), es decir aquellas evidencias que no revelan nada más que la validación afirmativa de una condición.

Aplicando estos principios funcionales a la analogía planteada anteriormente obtendríamos una instancia práctica del ecosistema necesario para una gestión soberana de la identidad.

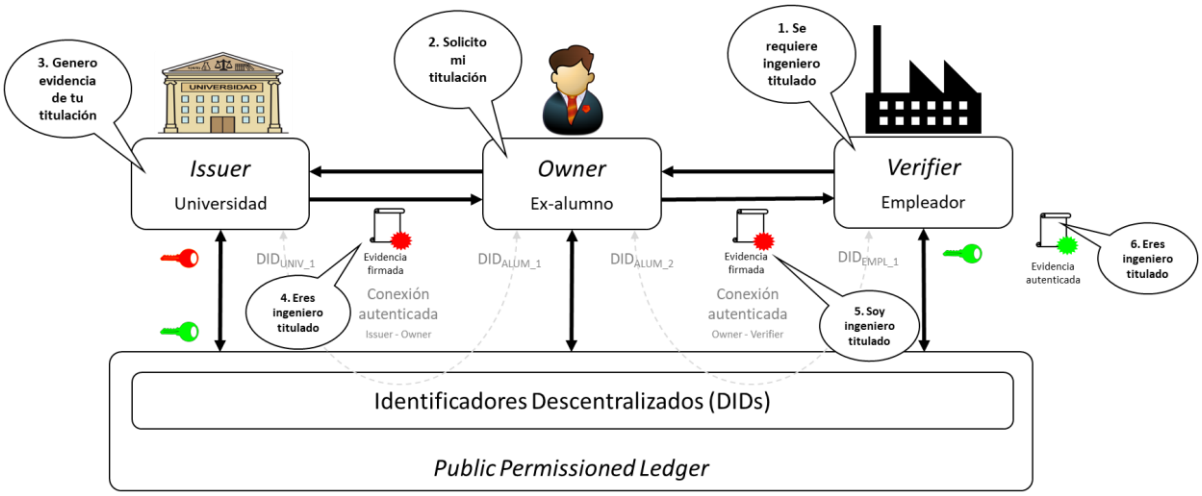

Ilustración 1. Ecosistema para la gestión soberana de la identidad

Como vemos, es el exalumno (*Owner*) quién decide qué parte de su identidad revela, y con quién la comparte, ya que ha establecido una conexión autenticada con el empleador (*Verifier*). Por ejemplo, si el empleador no nos ha requerido cuáles fueron las notas medias de la carrera, no será necesario que se las hagamos saber.

Por otra parte, el empleador no tendrá porqué confiar ciegamente en nuestro *curriculum vitae*, ya que dispondrá de una evidencia autenticada por nuestra universidad. Si el empleador confía en la universidad aceptará nuestra titulación, identificándonos únicamente como ingeniero titulado. No sería necesario que conociera ni nuestra edad, ni nuestro género, ni incluso nuestro nombre. Nosotros podríamos decidir si lo revelamos o no.

Del mismo modo podrían ser necesarias evidencias agregadas, dónde además de demostrar nuestra titulación universitaria fuera necesario demostrar nuestro lugar de residencia o nuestro conocimiento de idiomas. En ese caso podríamos decidir si creamos una evidencia combinada con la firma de la universidad, la del ayuntamiento de nuestra localidad y la de nuestra escuela de idiomas, o si por el contrario presentamos tres evidencias por separado.

Presentado el nuevo paradigma, su ecosistema, sus componentes clave y sus principios de funcionamiento es necesario analizar cuan madura está la tecnología, y qué herramientas tenemos a disposición para poder sacarle partido. De este modo podremos tangibilizar lo expuesto y profundizar en los detalles técnicos y funcionales, buscando una conclusión documentada acerca de la viabilidad de esta novedosa manera de gestionar nuestra identidad.

Así, en los próximos capítulos aparecerán las soluciones de mercado disponibles, las aplicaciones necesarias para implementarlas y las organizaciones que están detrás de ellas.

[página intencionadamente en blanco]

3. DESCRIPCIÓN DE LA SOLUCIÓN TÉCNICA: SOVRIN - IDENTITY FOR ALL

Cuando hablamos de gestión soberana de la identidad, varias son las iniciativas técnicas que se están poniendo en marcha, ya que se trata de un área relativamente reciente (2016). No obstante, una de estas iniciativas destaca a día de hoy por encima de las demás. Se trata de **Sovrin – Identity for All**.

3.1 LA SOVRIN FOUNDATION Y SU ORGANIZACIÓN

Antes de analizar técnicamente Sovrin es bueno recordar que, como hemos visto, la gestión soberana de la identidad no requiere únicamente de una herramienta concreta, sino de todo un ecosistema.

Esto es justamente lo que ofrece la ***Sovrin Foundation*** (https://sovrin.org/), una organización sin ánimo de lucro nacida de la colaboración entre varias empresas públicas y privadas de todo el mundo. Su misión es la de **proveer la infraestructura necesaria** para hacer accesible la gestión soberana de la identidad a cualquier persona u organización.

Una vez concebido el embrión de la solución técnica capaz de soportar el paradigma de la gestión soberana de la identidad, ha sido y es la *Sovrin Foundation* la encarga de gobernar su implantación y su adopción. Para ello la fundación se encarga de redactar y aprobar las especificaciones técnicas y funcionales, de definir y asignar los roles de los participantes, de establecer los derechos y obligaciones de cada una de las partes y de velar por el buen comportamiento de todos los asociados.

Para ello la *Sovrin Foundation* cuenta con 3 **órganos de gobierno** principales: el comité ejecutivo (*Executive Team*), el consejo de comisarios (*Board of Trustees*) y el consejo de gobernanza técnica (*Technical Governance Board*). El primer órgano está a cargo de la gestión operativa de la fundación, es decir de organizar el trabajo diario y de administrar sus cuentas. El segundo está a cargo de las decisiones vinculadas con la organización funcional del ecosistema y el tercero a cargo de tomar las decisiones técnicas vinculadas a la plataforma.

Veremos más detalle acerca de las funciones de cada uno de los órganos de gobierno una vez hayamos analizado el ecosistema desde el punto de vista tecnológico.

3.2 EL ENTORNO TECNOLÓGICO DE SOVRIN

Como hemos visto, pilar fundamental para el paradigma de la gestión soberana de la identidad es la tecnología *blockchain*, y más concretamente un modelo de blockchain del tipo *permissioned*, es decir, en el que solo los miembros autorizados pueden escribir nuevos bloques.

Desde el nacimiento del Bitcoin, varios han sido los proyectos que se han puesto en marcha para implementar entornos que permitirán trabajar con la lógica aplicada en las cadenas de bloques.

Uno de estos proyectos, hospedado por la *Linux Foundation* (https://www.linuxfoundation.org/), es el proyecto de código abierto Hyperledger (https://www.hyperledger.org), nacido a finales de 2015, y que busca a través de un marco técnico común (*framework*) potenciar el desarrollo de la tecnología *blockchain* aplicada a distintas áreas de negocio (banca, finanzas, IoT, manufactura y producción, etc.). Actualmente el **proyecto *Hyperledger*** cuenta con la colaboración de más de 250 compañías a nivel internacional, algunas de talla mundial como IBM®, SAP® o Intel®.

3.2.1 LA BLOCKCHAIN HYPERLEDGER INDY

El *framework* de *Hyperledger* ampara a su vez varios proyectos específicos, que desarrollan diferentes soluciones sobre *blockchain* según su finalidad. Uno de estos proyectos es el proyecto ***Hyperledger Indy*** (https://www.hyperledger.org/projects/hyperledger-indy), cuyo código original fue desarrollado por *Evernym* (https://www.evernym.com), posteriormente donado a la *Sovrin Foundation*.

La misión de este proyecto es la de **proveer un ecosistema técnico** para la gestión soberana de la identidad. El núcleo tecnológico de la *blockchain* de este proyecto es ***Indy Plenum***, un protocolo que regula el consenso entre nodos y la propagación de los bloques en base a un mecanismo conocido como **RBFT** o *Redundant Byzantine Fault Tolerance* (Aublin, Mokhtar, & Quéma, 2016).

Se trata de un protocolo operacional que contempla el liderazgo de un determinado nodo que actúa como primario, cuya actividad es monitorizada por el resto de nodos de la red (seguidores). En el caso de que el rendimiento del nodo primario sea identificado como deficiente o anómalo por el resto de nodos, estos inician un proceso que les permite tomar el papel de líder (primario).

En *Indy Plenum* los nodos se encargan de mantener actualizado el log de transacciones (en cadenas de bloques) en los denominados *ledgers*, y su propio conjunto de estados (estados de los nodos). Para esto último se utiliza la técnica criptográfica conocida como ***Merkle Patricia Trie***, utilizada también por la *blockchain Ethereum*, y que no es más que un mecanismo para agregar información en base a *hashes* de manera que esta sea fácilmente verificable en su conjunto.

Otra característica fundamental de *Indy Plenum* es el uso por parte de los nodos del protocolo técnico de comunicaciones conocido como ***CurveZMQ*** (http://curvezmq.org/), un protocolo de autenticación y encriptación basado en técnicas de criptografía de curva elíptica desarrollado por iMatix© (https://www.imatix.com), una empresa de software libre. Los nodos establecen entre si (y con el resto de actores) una conexión P2P mediante la cual, usando CurveZMQ como capa de seguridad, **intercambian ficheros en formato JSON**.

Finalmente, la persistencia en *Indy Plenum* se gestiona con **LevelDB** (http://leveldb.org/), una librería de código abierto para la escritura en disco de pares clave-valor (NoSQL).

Es sobre este ecosistema técnico *blockchain* (*Indy Plenum*) sobre el que se fundamenta a su vez el ecosistema funcional de Indy, que al tiempo es el ecosistema adoptado y caracterizado por Sovrin.

3.2.2 LA ARQUITECTURA DE SOVRIN

Después de un año de desarrollos sobre la plataforma *Hyperledger Indy*, la *Sovrin Foundation* realizó las primeras pruebas funcionales sobre lo que se denominó la ***Sovrin Network***, demostrándose en ese momento (mediados de 2017) la viabilidad técnica del ecosistema propuesto. A partir de ese momento, la *Hyperledger Indy* quedaría "fusionada" con la *Sovrin Network*, definiendo esta última al conjunto de componentes, y sus interrelaciones, necesarios para implementar el paradigma de gestión soberana de la identidad. Así, la *Sovrin Network* constituye la arquitectura de ecosistema vinculado a tal paradigma.

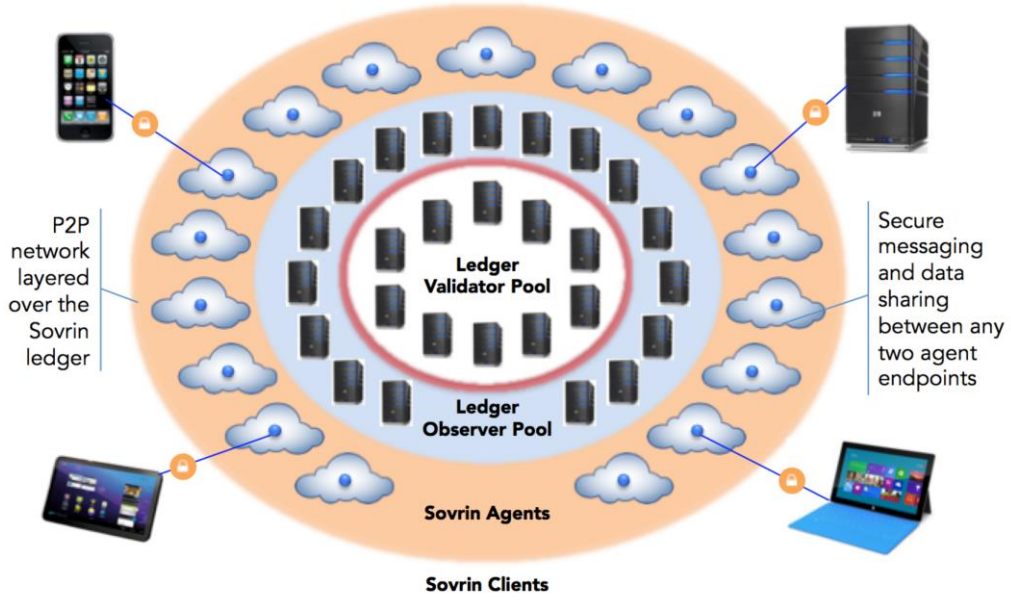

Ilustración 2. Arquitectura (Evernym, Sovrin Technical Foundations, 2016)

La arquitectura de Sovrin se divide en **4 capas fundamentales** (Evernym, Sovrin Technical Foundations, 2016).

La primera capa, la nuclear, está compuesta por nodos capaces de insertar nuevos bloques en el *ledger*, es decir, con **permisos de escritura**. Al tratarse de un *permissioned ledger*, estos nodos deben ser autorizados por los órganos de gobierno del sistema, aplicándose los procedimientos definidos. A estos nodos se les conoce como ***validator nodes***.

La segunda capa está compuesta por los llamados ***observer nodes***, que son nodos que únicamente tienen **permisos para leer** información del *ledger*. Puesto que las operaciones de escritura consumen muchos más recursos que las de lectura, la finalidad de los *observer nodes* es descargar a los *validator nodes*, resolviendo las consultas de información que lancen los clientes.

La tercera capa es la de los clientes que actúan de agentes o ***Agents***. Estos clientes efectivamente actúan como clientes de los nodos, pero a su vez como servidores de los clientes finales. Su función es la de actuar como un proxy para los clientes finales, ofreciéndoles **un punto de entrada fijo** a la *Sovrin Network*. Podemos verlo como un servicio en la nube al que acceden las aplicaciones que residen en los clientes finales, de manera que si un usuario utiliza varios (móvil, PC, tableta), estos puedan mantenerse sincronizados. Además, también ofrecen servicios de *backup* de información criptográfica y de almacenamiento seguro de información vinculada a la identidad.

Finalmente tenemos, en la capa más periférica, los **clientes finales**, que se ejecutan sobre los dispositivos terminales o *edge devices* (móvil, PC, tableta). Estos clientes son la interfaz directa con los usuarios, y se trata de aplicaciones que permiten relacionarse con la *Sovrin Network* de una manera usable e intuitiva. Los clientes acceden a la información contenida en los agentes, y solicitan las lecturas y escrituras en el *ledger* a través de estos, quienes acaban lanzando las peticiones a los nodos.

3.2.3 EL PAPEL DE LOS NODOS Y EL SOVRIN LEDGER

Los nodos forman parte de las capas nucleares de la *Sovrin Network*, y están a cargo de la **lectura y escritura del *ledger***, que es el registro (o base de datos) de información inmutable, inmutable gracias a la tecnología de cadena de bloques o *blockchain*. Pero ¿qué información contiene el *ledger*? Realmente no hay un único *ledger*, sino que hay cuatro, todos ellos necesarios para que el ecosistema técnico funcione.

El primero es el ***Identity ledger***, y es dónde se almacenan los denominados registros de identidad o ***identity records*** y los ***identifier records*** o registros de identificación, siendo estos últimos la base del sistema de gestión de claves públicas descentralizado o DPKI. Estos registros contienen los DID de cada uno de los participantes en el sistema.

Los DID son **identificadores descentralizados**, cuyo formato está definido y estandarizado por el W3C, y que básicamente sirven para identificar de manera inequívoca a los actores de la *Sovrin Network*, del mismo modo que las URL nos sirven para identificar de manera inequívoca a los recursos web de una red. Siguiendo con la analogía, igual que con una URL podemos acceder a una página web, con un DID podemos establecer una conexión con otro participante de la *Sovrin Network*.

La diferencia radica en que para conectarnos a la página web necesitamos de un servidor DNS que nos indique qué IP se corresponde con el nombre del servidor contenido en la URL. En este caso el servidor DNS actúa como entidad centralizada de confianza, ya que confiamos en la IP que nos facilita para establecer la conexión.

En la *Sovrin Network*, y gracias al *ledger* de DIDs, no es necesario esta figura de entidad centralizada de confianza para establecer una conexión entre dos actores. En este *ledger* se almacenan los DID y sus metadatos, en lo que se conoce como DID Document. Este documento contiene en un formato JSON todos los datos necesarios para identificar y autenticar a la contraparte, es decir, quién es el propietario del DID, en

qué dirección lógica podemos localizarle (la *service endpoint*) y cuáles son las claves públicas de verificación asociadas a dicho propietario.

De este modo, si queremos establecer una comunicación con otro actor de la *Sovrin Network* deberemos conocer su DID. Con esta información preguntaremos al *Identity ledger* por lo metadatos asociados y sabremos entonces en qué dirección localizarle. Esta dirección apuntará a un Agente, que actuará a modo de proxy poniéndonos en contacto con el cliente que representa al actor. Con la clave pública de verificación obtenida también del *Identity ledger* podremos asegurar que el actor con el que nos estamos poniendo en contacto es quién dice ser, ya que será el único poseedor de la clave privada emparejada con esa clave pública de verificación. A este proceso de establecimiento de comunicación entre dos actores se le conoce como ***onboarding***, y termina con una **conexión P2P securizada y autenticada** entre las partes.

El DID utilizado por cada uno de los actores está bajo el control de los propios actores, que deciden cuando y cuantos generan. La creación de un DID básicamente consiste en la generación de un par de claves público-privadas. Una parte de la clave pública, codificada en un formato específico, se utiliza como identificador, mientras que la clave privada queda únicamente en manos de su propietario (se almacena en su ***wallet***). Una vez creado el DID, su propietario debe publicarlo en el *Identity ledger*, junto con los metadatos necesarios. Es en este momento de publicación, y de escritura en el *ledger*, cuando aparece el concepto de **gestión de la confianza**, ya que, como hemos visto, no cualquiera puede ejercer de *validator node*. Más adelante veremos en detalle cómo se gestiona la confianza en la *Sovrin Network*, garantizándose que todo lo escrito en el *ledger* tiene un origen lícito y fiable, gracias a los ***Stewards*** y los ***Trust Anchors***. Por el momento identificaremos a estos roles como **garantes de la confianza**.

En lo referente a la creación de DID existen dos tipos bien diferenciados: los ***verinym*** y los ***pseudonym***. En principio cada usuario de la *Sovrin Network* tendrá un único DID *verinym* y un número indefinido de DIDs *pseudonym*. La primera vez que nos demos de alta en la red necesitaremos generar el DID *verinym*, para lo que será imprescindible la participación de un agente de confianza. A este proceso se le conoce como ***provisioning***, y lo que hace es dar de alta una nueva identidad en la red. Una vez dispongamos de un DID *verinym* registrado en el *Identity ledger*, podremos solicitar operaciones de escritura en el *ledger* y actuar de manera autónoma como *Issuer* o como *Verifier*. Para reforzar la confidencialidad y la privacidad se recomienda utilizar un DID *pseudonym* distinto para cada relación que establezcamos.

Como hemos visto, en el *Identity ledger* también almacenamos los ***Identity records***. Estos registros pueden ser de varios tipos, pero los más importantes sin duda, son los ***claim records***. Como sabemos los *claims* son las **evidencias de los atributos** que han sido validados por un *Issuer* de confianza, así que los *claims* son lo que acaba conformando nuestra identidad en la *Sovrin Network*.

Existen dos tipos fundamentales de *claims*: los públicos y los privados. Los públicos serán todos aquellos que no revelen información personal que permita identificarnos o PII (*Personally Identifiable*

Information). Los privados, los que si lo hagan. Además, los *claims* pueden almacenarse en diferentes formatos, tales como en **texto plano**, **encriptados** o **en forma de hash**, como una mera evidencia de la existencia del propio *claim.* También tenemos los ***claims* anónimos**, conocidos como *Zero Knowledge Proof* o ZKP.

Por razones de seguridad los *public claims* y los *claims* anónimos son los únicos que se almacenarán en el *Identity ledger*. Por el contrario, los *private claims*, es decir, los que contienen información privada, nunca se almacenarán en el *Identity ledger*, aun y estando encriptados. Estos deberán almacenarse siempre ***off-ledger***, en un medio seguro de almacenamiento, que bien podría ser un *ledger* privado (u otro medio).

Los *claim record*, así como el resto de *Identity records*, siempre que se escriban en el *Identity ledger* irán asociados al DID del propietario de la identidad, en este caso del *Owner* del *claim*, de manera que el *ledger* tendrá registrado a qué DID pertenece cada *claim*. ¿qué pasa entonces con los *claim* sensibles que se almacenan fuera del Sovrin *ledger*?

Justamente otro de los tipos de *Identity records* es el ***Receipt record*** (Evernym, Sovrin Glossary, 2016), que sirve para registrar que el *Identity Owner* ha registrado un *claim* fuera del *ledger* (*off-ledger*). Ese registro contiene el identificador del *claim* registrado en el medio de almacenamiento externo a Sovrin. Existe otro tipo más específico de *Receipt record* llamado ***Consent receipt record***, que registra en el *ledger* una evidencia de que el *identity Owner* ha compartido ese *claim* privado con otro actor (por ejemplo, con un *Verifier*).

Otros tipos importantes de *Identity records* son los ***Link contracts*** y los ***Reputation records***. Los primeros registran quién está compartiendo datos con quién, y con qué finalidad. Los segundos sirven para registrar eventos que afectan (positiva o negativamente) a la reputación de cualquiera de los actores de Sovrin.

Además del *Identity ledger* (que contiene, entre otros, *Identifier records* e *Identity records*) los nodos gestionan también un segundo *ledger* llamado ***Pool ledger***. En este se almacena un listado con todos los nodos disponibles, listado que los clientes consultan para saber con qué nodo pueden interactuar para acceder a la información del *Indentity ledger* (para su lectura o su escritura). Se trata de la lista de nodos permitidos, porque recordemos, que Sovrin se basa en una *public permissioned blockchain*, así que debe existir una lista de nodos autorizados. La selección de nodos permitidos, y que pueden actuar como *Validators* o como *Observers,* se hace mediante un proceso de votación, en el que los agentes de confianza son los que tienen derecho a voto.

El tercer *ledger* que gestionan los nodos es justamente el que almacena los votos de los agentes de confianza, y el que acaba determinando qué nodos tienen permisos y qué rol ejecuta cada uno (*Validator* o *Observer*). Este se conoce como ***Voting ledger***. También se registran votaciones de otra índole, como por ejemplo las que permiten añadir a un nodo para que opere activamente como *Observer* o como *Validator*.

El cuarto y último *ledger* gestionado por los nodos es el llamado ***Config ledger***. En este se registran los parámetros de configuración comunes para la operación del sistema. Hay parámetros como por

ejemplo la cantidad de votos necesarios para añadir a un nuevo agente de confianza. El conjunto de parámetros técnicos, así como sus valores, es definido por la *Sovrin Foundation*, a través de su consejo de gobernanza técnica (*Technical Governance Board*).

Cuando derivado del funcionamiento operativo de la *Sovrin Network* se genera una transacción que es necesario registrar en el sistema, esta se envía a un *validator node*, y una vez validada la transacción, es el propio nodo quién actualiza el *ledger* que corresponda según su naturaleza. Así, para los clientes, la existencia de 4 *ledgers* distintos **es completamente transparente**.

Como hemos visto, gracias a los *Identifier records* y los DIDs, el *ledger* es la base de la **infraestructura descentralizada de gestión de claves públicas** (DPKI), que lo que nos permite es autenticar a los participantes de la *Sovrin Network* sin necesidad de una entidad centralizada o CA (*Certification Authority*).

Gracias a la DPKI podemos establecer conexiones seguras P2P entre los participantes estando seguros de que estamos en contacto con quién pretendemos estarlo (y no con un impostor). Una vez establecida la conexión P2P podemos iniciar el intercambio de *claims* y evidencias. Recordemos que para que un *claim* tenga validez para un *Verifier*, este deberá estar firmado por un *Issuer* de confianza (con un cierto nivel de reputación), así que es fundamental que podamos asegurarnos de que la firma del *claim* la hace alguien que realmente es quien dice ser.

Del mismo modo, cuando compartimos la evidencia con el *Verifier*, queremos asegurarnos de que él, y nadie más que él, tiene acceso a los datos de la evidencia, ya que estos sí, son datos personales asociados a nuestra identidad.

Es entonces en el conjunto de evidencias, con la información contenida en cada una de ellas, dónde se almacena el conjunto de atributos que conforman nuestra identidad digital, y como hemos dicho, estas evidencias estarán en medios de almacenamiento seguros cuando contengan datos personales (PII).

3.2.4 EL PAPEL DE LOS AGENTES Y LAS AGENCIAS

Existen dos funciones clave para los Agentes de la *Sovrin Network* (Evernym, Sovrin Technical Foundations, 2016). La primera es la de ofrecer un **punto de conexión direccionable** para los clientes. Así como los nodos tienen una dirección fija y conocida, los clientes requieren también de una dirección que, además, sea independiente del dispositivo que se utilice o del proveedor de servicios de Internet (ISP) que se contrate.

Como hemos visto, en la *Sovrin Network* la identificación de un actor se realiza mediante los DIDs. Cada DID se asocia entonces a una dirección lógica fija llamada *Agent endpoint*. Como usuarios de la red Sovrin (*Owners*) dispondremos de un *agent endpoint* para cada DID que hayamos dado de alta en el *Identity ledger*. Como *Owner* también tenemos el **control total sobre nuestras direcciones** (*Agent endpoints*), puesto que estas están vinculadas a nuestros DIDs. Siendo así, podremos

albergar nuestro agente en cualquier dispositivo o contratar el servicio a cualquier proveedor, portando con nosotros nuestras direcciones.

La segunda funcionalidad clave que ofrecen los agentes de la *Sovrin Network* es la de **contenedores seguros de información**. Cuando hablamos de información, básicamente tenemos tres tipos: (a) los gráficos que mantienen **relaciones** (*graphs*), (b) la información asociada las **evidencias privadas** (*private claims*) y (c) la **información de respaldo** (*backup*) de nuestras claves.

Las relaciones en Sovrin (a) pueden hacer referencia a información de diversa índole: 1) la información relativa a **nuestras identidades**, 2) la información relativa a todas **nuestras conexiones** con los distintos actores de Sovrin, 3) la información relativa a **nuestra reputación** y 4) la información asociada los **datos y activos digitales** que queramos gestionar desde Sovrin (archivos, fotos, videos, etc.).

Si nos centramos en los gráficos, la información relativa a nuestra identidad (1), denominada ***Identity Graph***, no es más que el conjunto *Identity records* que comparten un mismo *identifier record,* es decir, los *claims* (*identity records*) que tenemos vinculados a un mismo DID (*identifier*). Sería lo mismo decir que el *Identity Graph* nos permite saber qué identidad hemos creado apara cada uno de nuestros avatares anónimos (*pseudonyms*).

La información asociada a nuestras conexiones (2) nos permite ver la relación entre nuestros DID y los DID de los diferentes actores con los que hemos interactuado. Recordemos que las conexiones entre actores se establecen en Sovrin mediante P2P, en donde cada una de las partes se identifica con un DID. Esta información (el emparejamiento de DIDs) se almacena en lo que se conoce como ***Relationship Graph***.

La información relativa a la reputación (3) no es más que una especialización de un *Relationship Graph*, en el que cada una de las relaciones es una declaración reputacional, es decir un apunte positivo o negativo de la reputación del actor al que la conexión hace referencia. Esta información se almacena en los ***Reputational Graph***, que se construyen a partir de los *reputation records*.

La información relativa a nuestros activos digitales (4) es el conjunto de archivos que tenemos vinculados a nuestra identidad y que queremos gestionar desde Sovrin. Un ejemplo podría ser una foto digital que hubiéramos utilizado en una evidencia que atestiguara nuestro aspecto físico.

Además de los gráficos, los Agentes almacenarán (b) todos aquellos *claims* que deban **gestionarse *off-ledger*** por contener información personal. Serán los *Issuers* los que, a la hora de generar los *claims*, en lugar de escribirlos en el *Identity ledger*, los escribirán directamente sobre el agente que corresponda (el que esté asociado a nuestro DID). Al mismo tiempo podrán escribir en el *Identity ledger* (público) un registro (*receipt record*) con el qué probarán que han hecho una transacción *off-ledger*, haciendo referencia al *claim* privado para que este pueda ser identificado y verificado.

Por último, los Agentes también podrán almacenar (c) **copias de seguridad** de lo referente a las **claves criptográficas**, que como

veremos se gestionan desde los clientes. Esto permite simplificar los procesos de recuperación de claves en caso de necesidad.

Así, serán los agentes los encargados de gestionar la información relativa a nuestras relaciones en Sovrin (relación de identidades y *claims*, relación entre actores, relación de reputaciones y relación de nuestros activos digitales), la información asociada a nuestros *claims* privados y los *backups* de nuestras claves. Toda esta información estará almacenada en un **contenedor lógico**, que el propio Agente se encargará de instanciar en uno o varios contenedores físicos debidamente securizados (*private ledger*, bases de datos encriptadas, servicios de almacenamiento *public cloud*, *private cloud*, *hybrid cloud*, etc.).

Vistas las funciones principales de un Agente en Sovrin, vemos que es necesario que estén albergados físicamente (*hosted*) en algún entorno. Podría ser el usuario (*Owner*) quién albergara al Agente en su propio equipo, pero debería garantizar que este está permanentemente conectado a la red para que estuviera disponible para todos sus clientes. Siendo así, el usuario, en lugar de instalarse y mantenerse su propio Agente, puede elegir el disfrutar del mismo **a modo de servicio**, igual que lo hace con una cuenta de email o con un repositorio de archivos. Son los proveedores de estos servicios, a los que se conoce como **Agencias**, los que ofrecen las funcionalidades de Agentes a los usuarios. Así, la calidad de los Agentes podrá variar en función de la Agencia que se elija. Podrán existir agencias que ofrezcan servicios de Agente de manera gratuita y podrán existir Agencias que cobren por ello.

Resumiendo, los Agentes son los **puntos de contacto permanente** entre los nodos (el *ledger*) y los clientes. Son el punto de sincronización entre los diferentes clientes que pueda tener un usuario y también son las áreas de almacenamiento privado de los usuarios. Las Agencias son las entidades que ofrecen las funcionalidades de Agentes a los usuarios, sin que estos necesiten hacerse cargo de su instalación y mantenimiento.

3.2.5 EL PAPEL DE LOS CLIENTES

Los **Clientes** son los elementos que permiten a los usuarios de Sovrin **relacionarse con el sistema** (Evernym, Sovrin Technical Foundations, 2016). No obstante, además de ofrecer la interfaz de usuario también están a cargo de dos funcionalidades clave: la **gestión de las claves criptográficas** y el mantenimiento de una **copia local de la información** asociada a la identidad del usuario.

Como sabemos, uno de los pilares de Sovrin es la infraestructura DPKI, que se basa en el uso de **criptografía asimétrica**. De un modo muy simplificado podríamos decir que la criptografía asimétrica es aquella que requiere claves distintas para encriptar y para desencriptar mensajes. Así, nosotros podremos encriptar un mensaje con una clave y permitir que este sea desencriptado con una clave distinta. La clave de desencriptación la podríamos hacer llegar a los destinatarios potenciales sin miedo a que estos suplantaran nuestra identidad o modificaran nuestro mensaje, ya que no tendrían capacidad para volver a encriptarlo; con su clave

únicamente lo podrían desencriptar. Al par de claves que encriptan y desencriptan se le conoce como **claves privada y pública**.

El mecanismo de autenticación basado en criptografía asimétrica consiste en encriptar un mensaje conocido y hacérselo llegar al destinatario que pretende autenticarnos (confirmar que somos quien decimos ser). Además del mensaje encriptado le hacemos llegar al destinatario nuestra clave pública. Como esa clave pública únicamente podrá desencriptar un mensaje encriptado con su correspondiente clave privada, si el destinatario puede descifrar el mensaje, estará 100% seguro de que este ha sido encriptado con la clave privada pareja a la clave pública recibida. Así el destinatario confirma que el mensaje ha sido encriptado (enviado y firmado) por el propietario de la clave pública que ha recibido.

En el ecosistema habitual de seguridad basado en claves públicas (PKI) existe una tercera entidad de confianza que, mediante un certificado, certifica que una clave pública determinada está asociada a un actor concreto. Está tercera entidad es la *Certification Authority* o CA. Se presume que todos los actores confían en la CA, por lo que, si esta certifica que una clave pública me pertenece, quién la reciba podrá estar seguro de que es mía. Si además con esa clave ha podido desencriptar el mensaje que ha recibido, podrá estar seguro de que he sido yo, y sólo yo, quién lo ha encriptado (lo he firmado).

Recordando lo ya citado anteriormente, la infraestructura de clave pública descentralizada de Sovrin (DPKI) prescinde de las CA y registra la propiedad de las claves públicas en el *ledger*. Así, en el *ledger* tenemos una relación de claves públicas con sus propietarios. No obstante, para mantener el anonimato, en lugar de hacer referencia a nuestro nombre, en el *ledger* se hace referencia a nuestro DID, por lo que tenemos una relación entre DIDs y sus claves públicas asociadas, y esta información puede ser consultada por cualquier actor en cualquier momento.

Claro está que el *ledger* sustituye a la tradicional CA, pero en Sovrin cada usuario sigue siendo **responsable de guardar la clave privada** asociada a cada una de sus claves públicas. Como en cualquier sistema basado en criptografía, si se pierde la clave privada la seguridad queda totalmente comprometida.

Esta es justamente una de las funciones fundamentales de los clientes en Sovrin, **almacenar y gestionar las claves privadas** de sus usuarios. Sabiendo que, además, como *Identity Owners* vamos a tener un DID distinto para cada conexión que establezcamos, y que cada DID tiene su par de claves (una pública anotada en el *ledger* y otra privada que debemos gestionar), vemos que la cantidad de claves privadas a administrar puede llegar a ser muy elevada. También cuando deseemos establecer una nueva conexión y tengamos que crear un nuevo DID, deberemos tener la capacidad de generar un par de claves nuevo, por lo que además de almacenar claves, el cliente debe **ser capaz de generarlas**. También de **reemplazarlas si es necesario**, ya que siempre podemos perder una clave privada o querer renovar una existente por cualquier otro motivo. Todo esto tiene que ver con la administración de claves, y es una funcionalidad imprescindible para un cliente de Sovrin. Más allá de las operaciones que se realicen sobre las claves, el lugar

donde los clientes las almacenan se conoce de manera genérica como monedero o ***wallet***.

El mantener una copia local de la información vinculada a nuestra identidad es la segunda función clave de los clientes en Sovrin. Hemos visto que nuestra información identitaria pública puede almacenarse en el *ledger*, y que los Agentes se encargan de almacenar de modo seguro nuestra información privada. No obstante, para que el sistema sea operativo y para que como usuarios tengamos una experiencia satisfactoria a la hora de utilizarlo, es necesario que no tengamos que depender exclusivamente de una buena conexión con los Agentes y de un ancho de banda determinado. Por ello los clientes deben ser capaces de **mantener localmente la información** y posteriormente sincronizarla con los Agentes. Además, deben garantizar que esta información se mantiene de manera completamente segura.

Siendo los clientes el punto de interacción de los usuarios, cuando un nuevo cliente quiere entrar en la *Sovrin Network* es necesario **darlo de alta**, para que, a partir de aquí, cada vez que interactúe pueda autenticarse debidamente y garantizarse así la integridad y seguridad del ecosistema. Así, antes incluso de poder utilizar los servicios prestados por los Agentes, los clientes deben realizar el proceso de (*first-time)* ***provisioning***, que involucrará a un agente de confianza, y que acabará por registrar nuestro DID y su par de claves correspondiente. El alta de clientes adicionales no requerirá de este proceso siempre que los asociemos al mismo DID.

3.2.6 LA GESTIÓN DE LA CONFIANZA

Vista la arquitectura de Sovrin y las funcionalidades de los componentes de cada una de sus capas, es necesario entender cómo se articulan los procesos de gestión que hacen que el sistema funcione. El componente principal a gestionar es la confianza, elemento clave en una red dedicada a mostrar y demostrar nuestra identidad a los demás.

En la *Sovrin Network* la confianza es un activo que se traspasa desde el núcleo del sistema hasta la periferia. La confianza es necesario depositarla en los actores que se van sumando a la red, básicamente en el momento en el que se dan de alta, que es cuando todavía son desconocidos en el ecosistema. Una vez registrados en el sistema, a los actores, siempre que se autentiquen con los mecanismos que hemos visto, se les supondrá confianza plena, ya que todas sus acciones vinculadas con la gestión de su identidad quedarán registradas de manera inmutable en el *ledger*. No obstante, para la primera vez que interactúan con la red es necesario que se les conceda de manera explícita esa confianza inicial.

Los encargados de transmitir esta confianza son los agentes de confianza, que además actúan siguiendo unas reglas definidas en un "contrato" global denominado ***Sovrin Provisional Trust Framework*** (Sovrin Foundation, Sovrin Provisional Trust Framework, 2017).

Este contrato define los distintos roles participantes en la red de transmisión y aseguramiento de la confianza (***Web of Trust***), así como las responsabilidades y las obligaciones de cada uno. Dicha cadena nace en

lo más alto con los ***Trustees***, pasa por los ***Stewards***, los ***Trust Anchors*** y finalmente acaba en los ***Identity Owners***.

Los *Stewards* son los actores confiables de facto, ya que son los encargados de operar los nodos que leen (*observer node*) y escriben (*validator node*) en el *ledger*. Recordemos que el *ledger* de Sovrin es del tipo *permissioned*, así que únicamente las entidades con permisos pueden interactuar con la *blockchain*. Los nodos de Sovrin empezaron siendo operados por las organizaciones fundadoras del sistema, estableciéndose estas como *Stewards* fundadores y teniendo obviamente, esa confianza de facto. A partir de ahí, y como prevén los mecanismos definidos por los órganos de gobierno de la *Sovrin Foudation*, los *Trustees,* en comité, son los encargados de autorizar la incorporación de nuevos *Stewards*. Cuando un nuevo *Steward* asume su rol es necesario que este firme un acuerdo (*Sovrin Steward Agreement*) que recoge explícitamente sus obligaciones y que le hace responsable incluso ante la Ley en caso de no cumplir con ellas. Este acuerdo viene definido en el marco del *Sovrin Trust Framework*.

El siguiente nivel en la cadena de confianza de Sovrin son los *Trust Anchors*, que son *Identity Owners* con un nivel reputacional suficiente como para presuponer que respetaran los propósitos, los principios y las políticas de la *Sovrin Foundation*. Estos deben superar unos requisitos determinados y comprometerse a cumplir con las obligaciones establecidas en el *Sovrin Trust Framework*. Los *Trust Anchor* tienen la potestad de dar de alta nuevos *Identity Owner* en la red (generar su DID *verinym*) y de invitarlos a establecerse como nuevos *Trust Anchors* (siempre que cumplan con los requisitos). Entre las obligaciones que tienen está la de asegurar que todos los *Identity Owners* que den de alta, hayan aceptado previamente los acuerdos que establece para ellos el *Sovrin Trust Framework*. Así, se hacen “responsables” de delegar la confianza en el último eslabón: los usuarios o *identity Owners*. Una vez asumen el rol de Trust Anchors, los usuarios de Sovrin pueden solicitar escrituras en el *ledger* y, por tanto, actuar de manera autónoma como *Issuers* o *Verifiers*.

Los *Identity Owners* están en el último nivel de la cadena de confianza, y para poder ser dados de alta en el sistema por un *Trust Anchor* es necesario que acepten el *Sovrin Identity Owner Agreement*, que básicamente les compromete a no abusar de la *Sovrin Network*, a no actuar de mala fe y a respetar el *Sovrin Trust Framework*. Se trata de una aceptación parecida a la que hacemos cuando aceptamos los términos de uso y licenciamiento de los productos software que habitualmente utilizamos. Con este compromiso la confianza llega hasta el usuario final (de la mano del *Trust Anchor*) y este puede empezar a utilizar la *Sovrin Network* y a crearse su propia identidad digital en base también, al buen hacer del resto de actores y participantes (*Issuers* y *Verifiers*).

Por último, remarcar que las Agencias e incluso los desarrolladores que se adhieran al Proyecto Sovrin o desarrollen productos compatibles con la *Sovrin Network*, también tienen obligaciones recogidas en el *Sovrin Trust Framework* y en sus acuerdos particulares (*Sovrin Agency Agreement* y *Sovrin Developer Agreement*).

3.2.7 LA GESTIÓN DE LAS EVIDENCIAS

Con la arquitectura, la gestión de la confianza y los roles principales claros, necesitamos ver como se resuelve técnicamente la gestión de las evidencias, que finalmente es lo que dota de su funcionalidad nuclear a la *Sovrin Network*. Tenemos claro que en el *ledger* se almacenan unos registros denominados ***claim records***, cuyo objetivo es afirmar uno o varios atributos vinculados a un *Identity Owner* (o a un DID concreto). El caso más recurrente es el de los registros que nos sirven para afirmar un conjunto de atributos (*credential)*.

Bien, para poder intercambiar los ***credentials*** (conjuntos de atributos "empaquetados") es necesario que tanto *Issuers* como *Verifiers* sean capaces de interpretar la información que contienen. Para ello se define su estructura semántica en lo que se conoce como ***credential schema***, que no es más que la definición de una "plantilla" en la que se indica qué atributos contendrá el *credential*, qué nombre le damos y qué versión, ya que puede ir evolucionando con el tiempo.

Por ejemplo, si necesitamos un *credential* que haga referencia a nuestro historial académico crearemos una "plantilla" a la que podríamos llamar "Referencia Curricular", con un identificador de versionado "1.0" y con un set de literales que definirán sus atributos que podrían ser "nombre", "apellidos", "titulación", "universidad", "carrera o grado", "año de titulación" y "nota media". Con esto habríamos generado el *credential schema* "Referencia Curricular". Este *schema*, por ejemplo, lo podría generar el Ministerio de Educación, a fin de estandarizarlo, y lo publicaría en el *ledger* para que cualquier actor pudiera utilizarlo. Las universidades, como *Issuers,* podrían ofrecerlo a sus estudiantes titulados. Los exalumnos como *Identity Owners* podrían solicitarlo a sus universidades. Y las empresas, como *Verifiers,* podrían solicitarlo a los candidatos. Todos ellos harían referencia al mismo *credential schema*. En cuanto a la creación de los *schemas*, son los *Trust Anchor* los que tienen capacidad de publicarlos en el *ledger*, así que si necesitáramos un *schema* ad-hoc, deberíamos contactar con uno de ellos.

Una vez definido el *schema*, los *Issuers* pueden hacer referencia al mismo para instanciarlo y convertirlo en un "documento" referido a su propia institución. Sería el equivalente a apropiarse de una "plantilla" genérica. A este proceso se le denomina ***credential definition setup***, y acaba generando un ***credential definition*** que solo puede ser publicado en el *ledger* por los actores con un rol de *Trust Anchors*. Una vez el Issuer dispone de un *credential definition*, el proceso habitual pasa por que estos se pongan en contacto con los *Identity Owners* y les ofrezcan la *credential definition* debidamente rellenada con sus datos y autenticada.

En nuestra analogía, la universidad nos contactaría y nos ofrecería un *credential definition* asociado al *schema* "Referencia curricular v1.0". Este *credential definition* ya sería propio de la universidad (por ejemplo "Referencia Curricular UOC v0.1"). Nosotros como exalumnos podríamos consultar en el *ledger* el contenido del *credential definition*, ya que su "plantilla" estaría publicada. Veríamos que esta contiene el nombre, los apellidos, la titulación, la universidad, la carrera o grado, el año de titulación y la nota media. Si nos interesara el *credential definition*

debidamente rellenado y firmado (siendo entonces un *claim* válido) necesitaríamos dar un paso importante, generar un ***master secret***.

El *master secret* nos sirve para que podamos demostrar que el *claim* ha sido emitido para nosotros, y solo para nosotros. Imaginemos que el *claim* no contiene los atributos “nombre” y “apellidos” o que no queremos hacerlos públicos. Es el *master secret* lo que demuestra es que el *Issuer* lo ha emitido para nosotros. Pero ¿cómo lo hace?

Siguiendo con el ejemplo, ahora tenemos una propuesta de *credential definition* que todavía no está firmada por el Issuer. Si nos interesa obtener el *claim*, en nuestra respuesta al *Issuer* incluiremos una clave privada (el m*aster secret*) que solo poseemos y conocemos nosotros. Cuando el *Issuer* firme el *claim* (el *credential definition*) incluirá como input este *master secret*, así que su firma estará vinculada a nosotros y solo a nosotros, porqué somos los únicos que poseemos esa clave (Sovrin Foundation, How Sovrin Works, 2016).

Con esto ya dispondremos de un *claim* que podremos utilizar para demostrar que nuestra universidad atestigua que hemos completado nuestra carrera. Además, también atestigua nuestro nombre y nuestros apellidos, cosa que podríamos utilizar en algún otro contexto. Realmente ahora disponemos de un conjunto de atributos verificados que son “nombre”, “apellidos”, “titulación”, “universidad”, “carrera o grado”, “año de titulación” y “nota media”.

Ahora, si un *Verifier,* por ejemplo, un empleador, solicitara evidencias (***proof request***) que demostraran ciertos atributos de los candidatos, lo haría aceptando únicamente ciertos *schemas* y/o *credential definition*. Podría solicitar un documento que evidenciara los atributos “nombre”, “apellidos”, “titulación” y “lugar de residencia”, pero que solo pidiera que estén certificados por un tercero (que sean verificables) los tres primeros (para el lugar de residencia no es necesario). En ese caso podríamos generar una evidencia (***proof***) con los atributos de nuestro *claim* de la universidad y hacérsela llegar al *Verifier*. Con ello el *Verifier* podría verificar que los atributos “nombre”, “apellidos” y “titulación” están atestiguados sin que la universidad participara de manera directa en el proceso y sin que tuviéramos que revelar todos los atributos de su *claim* (solo los que nos han solicitado en la *proof request*).

Como vemos, teniendo un conjunto de atributos verificados, podemos utilizarlos de manera individual cuando lo consideremos oportuno, disfrutando finalmente del **paradigma de gestión soberana de la identidad**, donde nosotros decidimos qué parte de nuestra identidad compartimos y con quién la compartimos.

Todos estos intercambios entre actores se realizan a través de los agentes, tal y como hemos visto en la arquitectura, y que, mediante comunicaciones P2P securizadas se produce un intercambio de ficheros JSON, siendo todo ello transparente para el usuario que está interactuando con el cliente instalado en su dispositivo.

3.2.8 LA GESTIÓN DE CLAVES

El último aspecto que queda por analizar del ecosistema de la *Sovrin Network* es la gestión de claves, elemento fundamental para su funcionamiento. Hemos visto que la autenticidad de los registros introducidos en el *ledger* se garantiza gracias a la criptografía asimétrica, y más concretamente gracias a las claves públicas que cada usuario pone a disposición de sus interlocutores.

Así, es fundamental en lo relativo a la gestión de claves, el poder obtener la clave pública que nos permita validar la autenticidad de un registro o de una conexión P2P. Al proceso de localizar estas claves se le denomina ***key discovery***, y básicamente consiste en consultar el *ledger* haciendo referencia a un DID determinado.

Los procesos de gestión de claves permiten también actualizarlas, es decir reemplazar las claves existentes por claves nuevas (el par pública-privada). Para ello primero debemos generar un nuevo par de calves y posteriormente debemos escribir en el *ledger* un nuevo registro con el DID que corresponda (el *verinym* o el *pseudonym* al que queramos actualizar las claves). Este registro lo firmaremos con la clave privada anterior, pero en él haremos constar la nueva clave. A partir de aquí todo el mundo que pregunte al *ledger* por nuestra clave, obtendrá como respuesta la nueva clave pública, por lo que ya podremos empezar a firmar con la nueva clave privada. En ese momento la clave anterior deja de tener validez. Al proceso de renovación de claves bajo circunstancias normales se le llama ***key rotation***.

Cuando las circunstancias no son normales, por ejemplo, cuándo nuestras claves privadas han sido robadas o comprometidas, al proceso se le conoce como ***key revocation***. Como hemos visto, para cambiar las claves necesitamos utilizar las claves antiguas (para realizar el cambio), por lo que si estas han sido comprometidas debemos evitar que un tercero pueda modificarlas. Para ello se pone en marcha el proceso de ***key recovery***.

Para iniciar este proceso es necesario que previamente hayamos designado a nuestros propios *trustees* (cualquier actor del sistema), que serán aquellos que nos ayudarán con la recuperación de las claves. Un número mínimo del total de estos *trustees* (en los que nosotros confiamos) tendrá potestad para firmar un registro asociado a nuestro DID. Si, por ejemplo, 5 de 7 *trustees* tienen esa capacidad, será necesario que al menos 5 de ellos (cualesquiera) se pongan de acuerdo para realizar dicho registro. Con esta técnica (secreto compartido) evitamos, o reducimos mucho el riesgo de que actúen contra nosotros de manera ilícita. Si los 5 colaboran atendiendo nuestra solicitud, los *validator nodes* validarán su nuevo registro después de un tiempo llamado ***timelock period***, asociado al proceso de *key recovery*. Este tiempo evita que el ladrón de claves pueda cambiar de manera inmediata nuestra clave anterior para asignarse la suya propia, garantizándose la seguridad del proceso.

3.3 EL ESTADO DEL ARTE DE LAS HERRAMIENTAS PARA SOVRIN

Hemos visto y analizado el entorno tecnológico de Sovrin, y hemos comprobado que realmente existe una solución tecnológica capaz de

soportar y de implementar el paradigma de gestión soberana de la identidad. Hemos visto también, que la implantación de dicho paradigma comporta una parte puramente tecnológica y otra parte más organizativa. Además, hemos comprobado cuan reciente es la propuesta y cuáles son sus orígenes. Todo ello hace que sea necesaria una revisión del estado del arte de este ecosistema (a noviembre de 2018).

Para empezar, deberíamos contextualizar el Proyecto Sovrin ubicándolo adecuadamente en la etapa de su ciclo de vida tecnológico. El Proyecto heredó la tecnología del Proyecto *Hyperledger Indy* a finales del año 2016, por lo que podríamos situar en esas fechas su nacimiento formal. Esto significa que se trata de una tecnología emergente, que todavía tiene que llegar a su fase de crecimiento y a su etapa de madurez. Además, aunque coordinado por la *Sovrin Foundation*, se trata de un Proyecto abierto que requiere de la contribución de la Comunidad de Internet, ya que aspira a establecerse como un estándar *de facto*. Esto hace que su evolución siga un ritmo mucho más lento si lo comparamos con cualquier iniciativa tecnológica puramente comercial.

Con todo esto podríamos decir que el ecosistema Sovrin se encuentra en construcción a día de hoy, lo que limita de manera muy significativa las herramientas a disposición de los usuarios. De hecho, no existen todavía soluciones con una madurez suficiente como para que los interesados puedan utilizar de manera efectiva la *Sovrin Network*. Lo que está en marcha son iniciativas orientadas a dar a conocer y a seguir desarrollando la tecnología, e iniciativas enfocadas en el consenso y el cierre de las políticas de gestión de la confianza (*Sovrin Gobernance Framework*).

En cuanto a la gobernanza y a las reglas que deberían regir en el ecosistema, se ha lanzado a revisión pública la segunda versión del set de documentos que componen el modelo de gobernanza. El periodo para que cualquier interesado pueda contribuir con sus aportaciones y comentarios se ha abierto desde el 1 hasta el 26 de noviembre de 2018. A partir de ahí los comentarios serán revisados por el grupo de trabajo dedicado a ello (*Sovrin Gobernance Framework Working Group*) y si procede, serán aprobados por el Consejo de la Fundación (*Sovrin Foundation Board of Trustees*). Esto puede situar el *release* del nuevo *framework* a mediados de 2019. Mientras tanto, y como hemos visto, el ecosistema se rige por la propuesta inicial, denominada *Sovrin Provisional Trust Framework*.

Esto hace que, obviamente, todavía ninguna organización dedicada al desarrollo de software se haya decidido a lanzar un producto comercial completamente funcional. Para ello tendremos que esperar a que el ecosistema madure un poco y se asiente en la comunidad, si realmente tiene aceptación.

No obstante, como siempre ocurre con las iniciativas abiertas que se apoyan en la Comunidad de Internet, lo que sí que han aparecido son herramienta enfocadas a los desarrolladores. El objetivo de estas herramientas es difundir el conocimiento existente, poner a prueba la tecnología y dar soporte inicial a los grupos que quieran contribuir en el desarrollo del Proyecto.

3.3.1 EL SDK Y LA LIBRERÍA LIBINDY

Si nos centramos en este tipo de herramientas, que son las únicas que están disponibles para analizar el ecosistema de manera práctica, vemos que tenemos dos fuentes principales. Tenemos las herramientas del Proyecto *Hyperledger Indy* y las propias del Proyecto Sovrin. Ambas comunidades todavía siguen trabajando en paralelo, aunque se prevé que finalmente sea Sovrin quien acabe concentrando todos los esfuerzos orientados a la gestión soberana de la identidad (ya que *Hyperledger* no dispone de esa capa organizativa). De todas maneras, por el momento, las herramientas técnicas de uno u otro proyecto son completamente homólogas. La manera de distribuirlas y de ponerlas a disposición de la Comunidad de Internet también es la misma: compartir el código fuente y su documentación asociada a través de la plataforma colaborativa de desarrollo GitHub (https://github.com).

Dentro de las opciones que tenemos podemos encontrar dos, claramente diferenciadas. La primera son las aplicaciones que permiten ejecutar un Cliente y conectarnos a la red de Hyperledger o de Sovrin (están disponibles la red de pruebas *Test* o *Sandbox* o la red Productiva *Live*). La otra opción es la de trabajar con un SDK (*Software Development Kit*) en lugar de con una aplicación cliente. Con esta segunda opción en lugar de tener una aplicación "cerrada" con la que únicamente podemos interactuar a base de comandos, tenemos el código fuente preparado para poder construir nuestra propia aplicación, y poder ver así que ocurre en sus "tripas". Por este motivo trabajaremos con esta segunda opción. Podemos encontrar la documentación y el código fuente del SDK (llamado Libindy) en el repositorio GitHub para *Hyperledger Indy* (Artemkaaas & varios, 2018):

https://github.com/hyperledger/indy-sdk/blob/master/doc/getting-started/getting-started.md

Antes de poder trabajar con Libindy necesitaremos poner en marcha la infraestructura necesaria para poder analizar el caso práctico que implementa. Esta infraestructura está compuesto por un *pool* de nodos al que nos conectaremos y el entorno de desarrollo integrado (IDE) necesario para analizar y ejecutar el código fuente.

3.3.2 EL ENTORNO DE DESARROLLO INTEGRADO (IDE)

La Libindy ofrece a la comunidad de desarrolladores distintos *wrappers*, que permiten llamar a las funciones básicas de la librería desde funciones más agregadas, facilitando las tareas de integración. Estos *wrappers* están a disposición en varios lenguajes de programación: .NET, iOS, Java, NodeJS y Python. En cualquiera de estos lenguajes tendremos un set de funciones que acaban llamando a las funciones originalmente desarrolladas en la librería (implementada en C). En nuestro caso utilizaremos el *wrapper* de Python, por lo que necesitaremos un IDE para Python, que será PyCharm.

Una vez instalado PyCharm y descargada Libindy de GitHub, procedemos a abrir un proyecto con los módulos descargados (los del *wrapper* para Python). Después instalamos el paquete para Python indicado en los propios archivos (*indy_sdk-0.0.1*), que es donde están

implementadas las funciones primitivas en C. Entre los ficheros que integran nuestro proyecto tenemos uno llamado “getting_started.py”, que es el que contiene el programa principal que ejecuta la secuencia del entorno práctico. Este será el módulo que ejecutaremos para ver el funcionamiento del programa. No obstante, si intentamos hacerlo vemos que se produce un error, que nos indica que no se ha podido contactar con el *pool* de nodos.

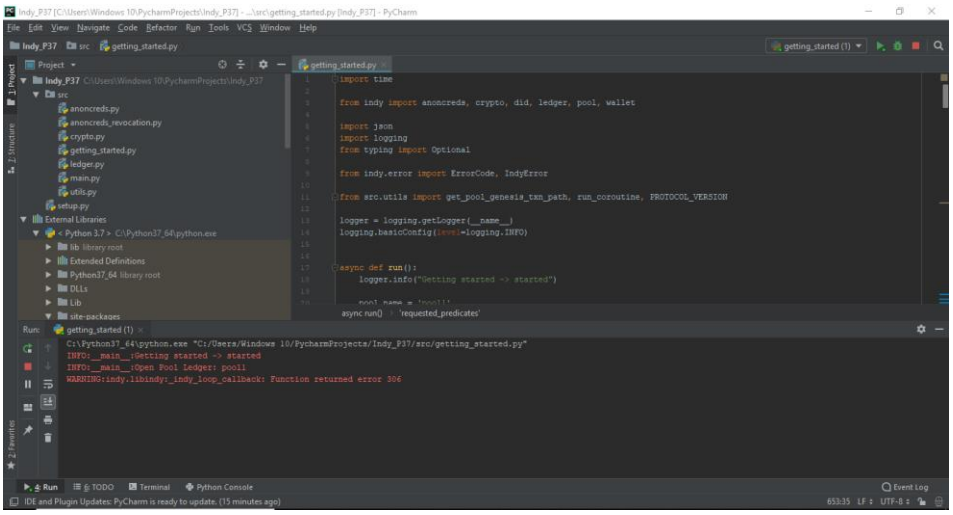

Ilustración 3. Módulo getting_started del wrapper para Python de Libindy

Para resolver este problema necesitaremos desplegar un *pool* de nodos ficticio en nuestro equipo, al que asignaremos una dirección IP a través de la cual podremos contactarlo.

3.3.3 EL ENTORNO PARA EL POOL DE NODOS

Para implementar el *pool* local de nodos necesitaremos construir una red virtual de *Indy nodes*. Para ello la Comunidad de desarrolladores pone a nuestra disposición una máquina virtual. Más que una máquina virtual se trata de un contenedor Docker. Este contenedor, además del sistema operativo sobre el que se ejecuta la aplicación contiene todos los componentes que esta necesita para funcionar, resolviendo todas sus dependencias. Así, instalando el contenedor, tendremos todo lo necesario para que el *pool* de nodos funcione. No obstante, el aplicativo que gestiona los contenedores (Docker) está originalmente desarrollado en Linux, por lo que necesitaremos también una máquina virtual Linux para ejecutar Docker, que a su vez cargará el contenedor con el *pool* de nodos (que también va sobre Linux).

Una vez hemos instalado Docker, y la máquina virtual que necesita, podemos descargarnos el contenedor del *pool* de nodos Indy. Esto podemos hacerlo desde la interfaz gráfica que ofrece Docker (Kitematic) o directamente desde la línea de comandos de nuestro sistema operativo (Windows 10 en este caso).

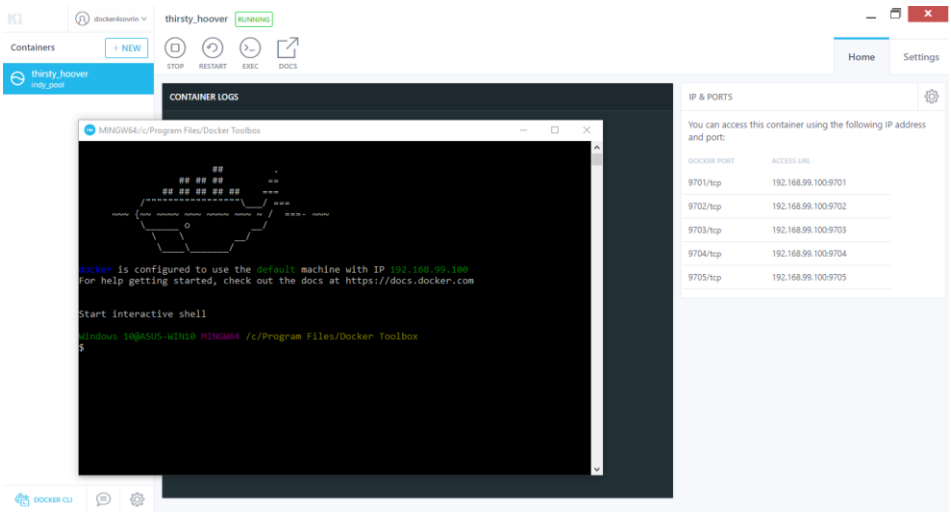

Ilustración 4. Contenedor Docker del pool de nodos Indy ejecutándose

A la hora de instalar el contenedor definiremos por qué dirección IP la aplicación tendrá conexión con nuestro sistema operativo (host). En nuestro caso hemos asignado la IP 192.168.99.100.

No obstante, recordemos que Docker está a su vez ejecutándose sobre una máquina virtual Linux, así que tendremos que configurar esta máquina virtual para redirigir el tráfico desde nuestro sistema operativo. Viendo que Libindy apunta a la IP 127.0.0.1 (localhost) y a los puertos 9701~9705 para conectarse al pool de nodos, deberemos redirigir el tráfico desde estos puertos hasta los puertos del contenedor. Para hacerlo configuramos el reenvío de puertos en nuestro gestor de máquinas virtual (VirtualBox).

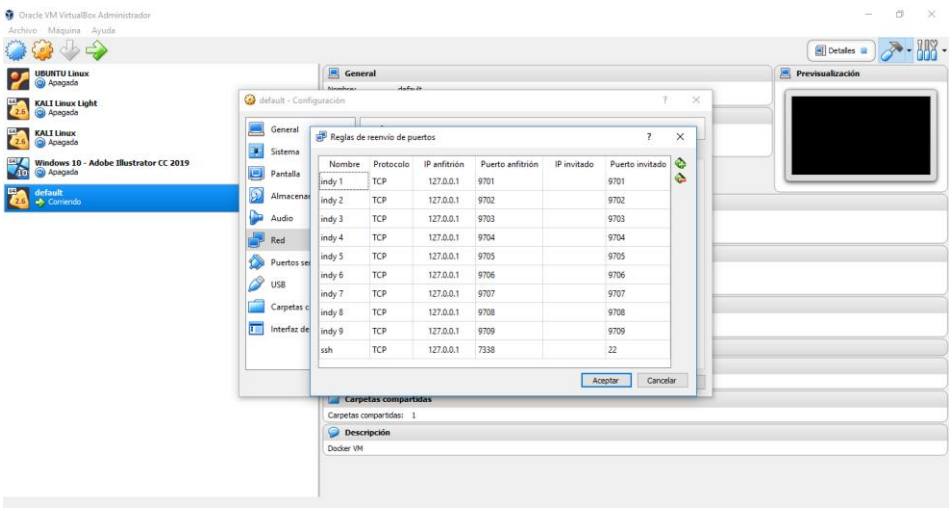

Ilustración 5. Configuración de puertos en la máquina virtual para Docker

Con esto ya podemos conectarnos con el pool de nodos desde nuestro sistema operativo, y más concretamente desde el *wrapper* para Python de la librería Libindy que estamos ejecutando en nuestro entorno de desarrollo (IDE).

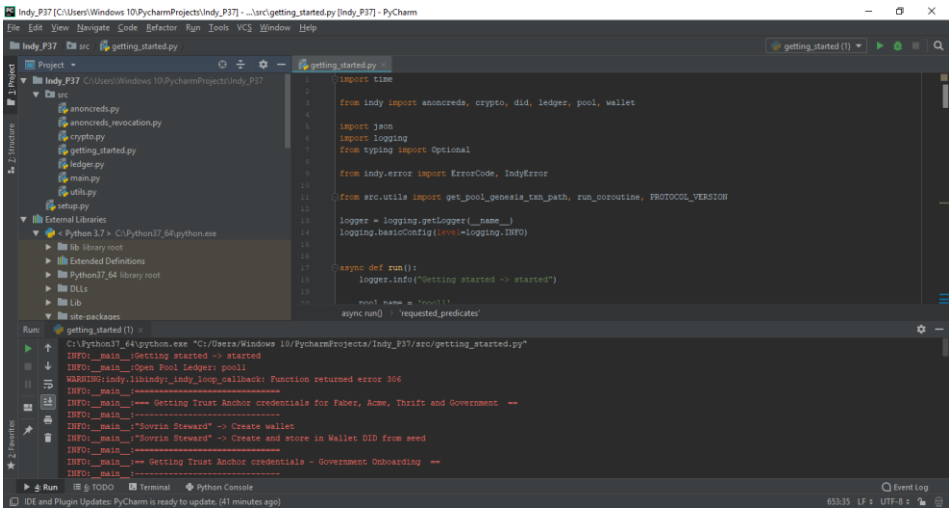

Ilustración 6. Conexión con el pool de nodos desde el entorno de desarrollo

Finalmente, para conseguir un entorno operativo que nos permita ejecutar el código del SDK hemos tenido que montar un *stack* tecnológico que queda de la siguiente manera:

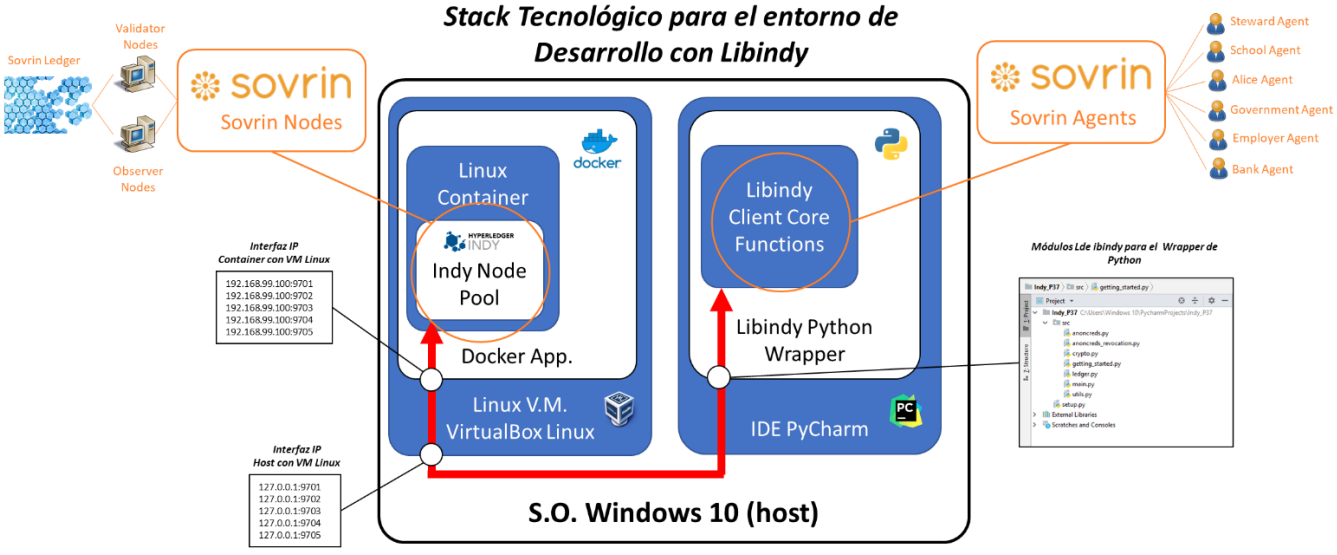

Ilustración 7. Stack tecnológico para el entorno de desarrollo con Libindy

Con el entorno de desarrollo ya operativo y teniendo clara su configuración, podemos pasar a analizar los módulos y las funciones principales del SDK Libindy.

3.3.4 LA ESTRUCTURA DE LIBINDY

Como hemos visto disponemos de un *wrapper* para Python de la librería Libindy. Este *wrapper* está compuesto por varios módulos que ponen a disposición de los desarrolladores un conjunto de funciones. Además, tenemos un módulo de ejemplo en el que se llaman a las funciones necesarias para implementar un caso de uso concreto. Este será el caso de uso que utilizaremos en nuestro entorno práctico para acabar de comprender como funciona la *Sovrin Network*. También será el módulo que desgranaremos a la hora de analizar las funciones principales del *wrapper*.

El conjunto de la librería (su *wrapper* para Python) queda conformado por los siguientes componentes, que no dejan de servir de ejemplo para los desarrolladores.

- **anoncreds.py**: en el que se implementan las funciones necesarias para que los *Issuers*, los *Verifiers* y los *Identity Owners* puedan interactuar con el sistema (crear credenciales, revocar credenciales, generar el *master secret*, obtener credenciales, buscarlas, generar esquemas, etc.)

- **crypto.py**: en el que se implementan las funciones necesarias gestionar la criptografía asociada al sistema. Hay funciones para generar pares de claves público-privadas, para firmar mensajes, y para desencriptarlos y autenticar al remitente.

- **did.py**: donde están definidas todas las funciones necesarias para gestionar los DID y para establecer conexiones entre ellos. Para la gestión de los DID es necesaria también la gestión de claves, así que se incluyen también las funciones para administrar tanto las claves públicas como las privadas (no para crearlas).

- **getting_started.py**: en el que se resuelve el caso de uso que analizaremos en el siguiente capítulo (4. Diseño del Caso de Uso para el entorno Práctico), y donde se ven las principales funcionalidades citadas a lo largo del apartado anterior (3.2 El entorno tecnológico de Sovrin). Las funciones que contiene este módulo son:

    - **onboarding**: que resuelve el proceso necesario para poner en contacto a dos actores en Sovrin. Este proceso consiste en generar un DID específico para la conexión (*pseudonym*), en hacérselo llegar a la contraparte y en nosotros verificar el DID recibido consultando su clave pública asociada en el *ledger* (la contraparte también autentica nuestro DID siguiendo el mismo proceso).

    - **get_verinym**: nos permite obtener el DID particular de un actor (*verinym*) y verificar su autenticidad obteniendo del *ledger* su clave pública correspondiente. Lo utilizaremos

principalmente para autenticar a los *Trust Anchors* y ver que realmente lo son.

- **send_nym**: para generar un *identity registrer* y solicitar a los *validator nodes* que lo den de alta en el *ledger*.
- **send_schema**: para dar de alta un *schema* en el *ledger* de manera que el resto de los actores lo tengan a disposición.
- **send_cred_def**: para crear y dar de alta en el *ledger* un *credential definition* en el que un *Issuer* instancia un schema genérico.
- **get_schema**: para obtener un *schema* del *ledger* y poder rellenarlo de manera que obtengamos un *claim* con un formato específico.
- **get_cred_def**: para obtener un *cedential definition*, es decir, un *schema* instanciado por un *Issuer* concreto. Se solicita al *ledger* para verificar qué atributos se certifican en ese *credential definition*.
- **get_credential_for_referent**: que sirve para que el *Identity Owner* busque evidencias entre sus *claims* que cumplan con los requisitos del *Verifier*, y pueda confeccionar su *proof*.
- **prover_get_entities_from_ledger**: para que el *prover* (el *Identity Owner*) pueda obtener del *ledger* los *schema* y los *credential definition* que necesite para ver qué atributos atestiguan.
- **verifier_get_entities_from_ledger**: para que el *Verifier* pueda obtener del *ledger* los *schema* y los *credential definition* que necesite para ver qué atributos atestiguan.
- **auth_decrypt**: que sirve para (1) desencriptar los mensajes recibidos y (2) autenticar a su remitente.

- **ledger.py**: donde se implementan todas las funciones necesarias para que el cliente Sovrin interactúe con el *ledger* que gestionan los nodos. Todo lo que tiene que ver con las funcionalidades que necesitan leer o escribir registros en los distintos *ledger* se resuelve en este módulo (crear *schemas*, recuperar *schemas*, escribir *Identity records*, obtener credenciales, etc.).

- **pool.py**: donde están definidas las funciones que permiten al cliente Sovrin crear un fichero de configuración del *pool* de nodos, abrir una conexión contra ese *pool*, refrescar la copia local de conexiones con los nodos de un *pool*, cerrar la conexión establecida, eliminar el fichero de configuración o definir la versión del protocolo de comunicaciones con los nodos.

- **utils.py**: en el que se implementan varias funciones que a su vez son requeridas y utilizadas por el resto de los módulos. Son funciones que básicamente sirven para gestionar la persistencia de la información en el cliente (dónde se guarda el wallet, cómo se actualiza su información, etc.). También configuran la IP en la que ir a buscar el *pool* de nodos (en nuestro caso 127.0.0.1).
- **wallet.py**: en el que se implementan las funciones necesarias para gestionar los *wallets*. Estas funciones permiten crear un *wallet* en un cliente, abrir uno existente, cerrarlo, eliminarlo o importarlo y exportarlo entre distintos clientes.

Con esto hemos visto todo lo referente al ecosistema Sovrin, desde cómo se organiza su Fundación, hasta su arquitectura, sus elementos funcionales clave, sus roles, la gestión de las evidencias, de la confianza, de las claves y las principales herramientas que a día de hoy están disponibles para la comunidad de desarrolladores.

[página intencionadamente en blanco]

4. DISEÑO DEL CASO DE USO PARA EL ENTORNO PRÁCTICO

Explicado el funcionamiento teórico de la *Sovrin Network*, y para asentar todos sus conceptos, se define en adelante un caso de uso que será resuelto en el entorno práctico. En este capítulo se identificarán los actores involucrados, las necesidades de cada uno y las interacciones previstas para resolverlas en el ecosistema Sovrin (Sovrin Foundation, How Sovrin Works, 2016).

El caso de uso expuesto es el que se resuelve en la librería Libindy (Artemkaaas & varios, 2018), ya que la falta de herramientas comerciales hace que debamos utilizar aquellas disponibles para la comunidad de desarrolladores. Como tampoco es la finalidad de este trabajo el bajar a nivel de programación, utilizaremos el caso ya implementado, analizando paso a paso su funcionamiento, de manera que podamos vincularlo con los conceptos teóricos hasta ahora expuestos. Se respetarán los mismos actores e incluso los mismos nombres para facilitar al lector la asociación del presente trabajo con lo publicado en la web (Artemkaaas & varios, 2018) y lo escrito en el código fuente.

4.1 EL CASO DE USO: LOS ACTORES IMPLICADOS

El caso de uso gira entorno a Alice, que será nuestro *Identity Owner*. Alice necesitará interactuar con su universidad para obtener un certificado que a su vez le permita aplicar a un puesto de trabajo. Una vez obtenga el puesto de trabajo solicitará a su empleador un certificado que le permita pedir un préstamo al banco para poder comprarse un coche. Para resolver el caso será necesaria la participación de otros actores, como pueden ser los *Stewards* o una organización gubernamental que estandarice el formato de las evidencias. Presentamos a todos estos actores a continuación.

4.1.1 ALICE: EL IDENTITY OWNER

Alice es una exalumna que acaba de finalizar su carrera y que pretende aplicar a un puesto de trabajo en el que solicitan una cualificación equivalente a la suya. Alice no es usuaria de Sovrin, pero ha visto que el empleador le solicita demostrar de manera fehaciente su nivel académico. Además, ha recibido también un mail de su universidad en el que se le ofrece un “certificado” digital a modo de documento oficial mientras se tramita la expedición formal de su título. A raíz de esto se ha interesado por la *Sovrin Network*.

4.1.2 FABER: EL ISSUER

Faber es la universidad en la que Alice ha cursado sus estudios superiores. Es una universidad que intenta estar al día de los avances tecnológicos, y que trata de acompasarse con las necesidades de la industria, a fin de facilitar a sus alumnos la inserción en el mercado laboral. Faber quiere mejorar y optimizar sus procesos internos y ha puesto el foco

en la gestión del ciclo de vida asociado a la expedición y entrega de las titulaciones. El actual sistema de emisión de un certificado provisional en un formato no digital (de papel) no es demasiado cómodo ni para la propia universidad ni para los alumnos. Después de varios contactos con el Ministerio de Educación se ha suscrito a un proyecto piloto que pretende digitalizar las evidencias curriculares de los estudiantes. El proyecto se fundamenta en el ecosistema Sovrin y en la gestión soberana de la identidad. Convencidos del potencial de este proyecto y del impacto positivo sobre sus procesos de gestión han decidido formar parte del mismo, y con la ayuda del Gobierno han buscado una agencia que les ofrece las funcionalidades necesarias (las de *Agent*) para ofrecer el servicio a sus alumnos. Después de poner en marcha la infraestructura tecnológica han iniciado una campaña de comunicación con sus exalumnos.

4.1.3 GOVERNMENT: EL PROPIETARIO DEL SCHEMA

El Gobierno, a través de varios de sus ministerios, es quien ha lanzado un proyecto piloto para ayudar a las universidades a mejorar el proceso de emisión de titulaciones universitarias. Además, el proyecto también pretende reducir el fraude relativo al "inflado" de los curriculum y luchar contra los abusos en lo referente a la solicitud de información personal que se dan en los procesos de selección de empleados y en la concesión de financiación por parte del sector bancario. Por todo ello ha involucrado a universidades, a empresas y a los principales actores del entorno financiero para poner en marcha la infraestructura necesaria para trabajar con evidencias digitales en un ecosistema que permita la gestión soberana de la identidad. Después de varios meses de prospecciones se ha decantado por colaborar con la *Sovrin Foundation*. Como gobierno subvencionará la implantación de este ecosistema a universidades, empresas y bancos, ofreciéndoles además todo el asesoramiento técnico que precisen. Como ente regulador y "estandarizador", el gobierno será quien, a través del Ministerio de Educación y del Ministerio de Empleo, defina el formato de las evidencias digitales (*credential schema setup*), tanto en lo referente al historial académico (*Transcript*) como en lo referente al historial laboral (*Job-Certificate*).

4.1.4 ACME: EL PRIMER VERIFIER

ACME es una empresa tecnológica que ofrece servicios de asesoramiento y de gestión digital de la información a entidades públicas. Por ese motivo ha estado atenta desde el primer día al proyecto piloto que el Gobierno quiere poner en marcha, relativo a la digitalización de evidencias curriculares y a la gestión soberana de la identidad. Desde ACME confían además en que la adopción de este nuevo ecosistema les ayudará a optimizar sus procesos de reclutamiento de empleados, sobre todo en lo referente a la gestión de datos personales y a la verificación de las referencias facilitadas en los curriculum de los candidatos. En coordinación con el Gobierno, y acorde a la actual etapa del proyecto piloto, ha decidido publicar ya sus próximas vacantes exigiendo una evidencia digital acorde al *framework* de Sovrin. También ofrecerá a sus

empleados una evidencia digital que les permita demostrar que disponen de un empleo y de un número de afiliación a la Seguridad Social.

4.1.5 THRIFT BANK: EL SEGUNDO VERIFIER

Thrift Bank es un banco de tamaño medio, con un espíritu digital y cuyos ingresos provienen fundamentalmente del crédito minorista. No tiene músculo suficiente como para competir con los grandes bancos en grandes operaciones, pero está ganando cuota de mercado entre los *millennials* gracias a sus innovadoras políticas en cuanto a la gestión de datos personales y gracias a la plataforma tecnológica que pone a disposición de sus clientes. Acorde con su estrategia, ha decidido formar parte del proyecto piloto lanzado por el Gobierno. Considera que la participación en un entorno de gestión soberana de la identidad como el que ofrece Sovrin le ayudará a mejorar sus procesos de KYC (*Know Your Customer*) y de AML (A*nti-Money Laundering* o prevención de blanqueo de capitales). Además de mejorar la calidad y la fiabilidad de los datos de sus clientes, este ecosistema también le permitirá reducir los datos solicitados, quedándose únicamente con aquella información que realmente es necesaria para la concesión de financiación y para la gestión de riegos. El disponer de menos información personal reducirá los costes asociados a mantener dicha información acorde a la nueva regulación europea GDPR. Dada su experiencia, al banco no le ha sido difícil poner en marcha la infraestructura necesaria para empezar a operar bajo este nuevo ecosistema tecnológico, por lo que para la concesión de créditos al consumo ya está empezando a ofrecer a sus clientes la posibilidad de trabajar con evidencias digitales sobre la plataforma Sovrin.

4.1.6 STEWARD: EL GESTOR DE LA CONFIANZA

Contactada por el Gobierno, la *Sovrin Foundation* ha decidido implicarse de lleno en el proyecto piloto que este ha planteado. Para ello ha proporcionado todo el asesoramiento técnico necesario, y además ha nombrado a un *Steward* que monitorizará de manera permanente todo el proyecto. Se trata de una de las empresas fundadoras de Sovrin, que además opera varias decenas de nodos. No obstante, en el proyecto su papel fundamental será el de *Trust Anchor*, rol que se da por supuesto a cualquier *Steward*. Participará fundamentalmente en el alta de los distintos actores en el *Sovrin Ledger*, y velará por el respeto al *Sovrin Trust Framework* por parte de todos los participantes.

4.2 EL CASO DE USO: LA SECUENCIA DE ACONTECIMIENTOS

Presentados todos los actores y sus papeles en el caso de uso, definiremos ahora la secuencia de acontecimientos que caracterizaran nuestro caso práctico.

4.2.1 DANDO DE ALTA A LOS ACTORES EN LA SOVRIN NETWORK

El primer paso para poner en marcha el ecosistema de la *Sovrin Network* será el dar de alta a todos los actores. Para darlos de alta será

necesario que establezcan primero una conexión con el *Steward* de manera que ambas partes queden autenticadas y dispongan de un DID (*pseudonym*) para comunicarse. Una vez establecida una relación con el Steward cada actor podrá generar un nuevo DID y pedir al *Steward* que, en su rol de *Trust Anchor*, lo dé de alta en el *ledger* como identificador único (*verinym*). Con esto los actores ya existirán en el *ledger* y podrán emitir evidencias (como *Issuers*), solicitarlas (como *Verifiers*) y definir sus *schemas* (como *Trust Anchor*). En el proceso de alta, el Steward asignará el rol de *Trust Anchor* a todos los actores, de manera que en adelante estos puedan realizar todas las acciones necesarias si tener que volver a recurrir a él. Para ello, cuando el Steward solicita el alta del DID (verinym) en el *ledger*, indica que el rol de su propietario es el de *Trust Anchor*. Esto no es necesario para Alice, ya que su papel será el de *Identity Owner*, por lo que ella podrá operar y relacionarse con el resto de actores utilizando DIDs anónimos (*pseudonyms*).

Así, el *Steward* llevará a cabo el proceso de *onboarding* (para establecer una conexión segura en base a *pseudonyms*) y el proceso de alta en el *ledger* (generación de *verinym* y asignación del rol de *Trust Anchor*) con el Gobierno, con la universidad Faber, con la empresa ACME y con el banco Thrift Bank.

4.2.2 ESTANDARIZANDO EL CONTENIDO DE LAS EVIDENCIAS

Ya con el rol de *Trust Anchor* asignado y con un DID *verinym* en el *ledger*, el Gobierno ya tiene capacidad para generar los *schemas* de las evidencias que servirán para atestiguar el historial académico y la posesión de un empleo. Para ello generará el *schema* llamado *Transcript* y el *schema* llamado *Job-Certificate*.

El esquema *Transcript* (v1.2) servirá para que las universidades emitan una evidencia conforme sus estudiantes han cursado con éxito un grado universitario. Este esquema incluye los atributos “Nombre”, “Apellidos”, “Grado”, “Estado” (en curso o finalizado), “Año”, “Nota Media” y “Número de Seguridad Social”.

El esquema *Job-Certificate* (v0.2) servirá para que las empresas emitan un certificado a sus empleados atestiguando que tienen un trabajo. Los atributos de este esquema son “Nombre”, “Apellidos”, “Salario” (bruto mensual), “Estado del empleado” (contrato temporal o fijo) y “Experiencia” (en meses).

Estos esquemas serán de dominio público (quedará publicados en el *ledger*), y cualquier universidad o cualquier empresa podrá utilizarlos para emitir credenciales de sus alumnos o empleados.

4.2.3 OBTENIENDO EL CERTIFICADO DE ESTUDIOS

Ahora es el interés de Faber el emitir una evidencia que demuestre que Alice ha completado con éxito sus estudios. Lo primero que tiene que hacer es publicar en el *ledger* un *Credential Definition* basado en el *schema* publicado por el Gobierno. Para ello recuperará del *ledger* el esquema *Transcript* (v1.2) y lo firmará con su clave privada asociada a su DID *verinym*. Una vez firmado lo volverá a publicar en el *ledger* y ya

tendremos el *Credential Definition* para el *Trasncript Credential Schema* publicado por Faber (le llamaremos *Faber Transcript*).

Una vez publicado el *Faber Transcript*, Faber se pone en contacto con Alice y ambas partes se autentican usando DID anónimos (*pseudonyms*). Una vez autenticados Faber hace llegar a Alice una propuesta de evidencia, que es el *Faber Transcript*. Alice consulta en el *ledger* los atributos que componen este *claim* y decide que le es válido y que le interesa.

Para que Faber pueda emitírselo a su nombre, necesita que Alice le haga llegar su *master secret*. Para ello Alice se "baja" del *ledger* el *Faber Transcript* y le hace llegar a Faber una solicitud de credenciales (*Credential Request*) que contiene una referencia a su *master secret*. Con esta solicitud recibida Faber ya puede emitir las credenciales a nombre de Alice. Para ello completa los atributos de Alice y le hace llegar el *Faber Transcript*. Alice lo recibe, lo autentica y lo guarda en su *wallet*, por lo que ahora ya dispone de una evidencia certificada que demuestra su nombre, sus apellidos, su titulación académica, su nota media y su número de la seguridad social.

4.2.4 APLICANDO A LA VACANTE LABORAL

Para poder aplicar a la vacante laboral de ACME, Alice se pone en contacto con la empresa. Ambas partes se autentican y establecen conexión mediante DIDs específicos (*pseudonyms*). Una vez conectados ACME hace llegar a Alice una solicitud de evidencias (*proof request*). Llamaremos a esta solicitud *Job-Application Proof Request* (v0.1), y ACME la emite en calidad de *Verifier*.

Alice examina la petición de ACME. Esta le solicita a Alice su nombre, sus apellidos, su titulación, el estado de su titulación, su número de la seguridad social, su número de teléfono y una nota media mayor o igual a 4.

Alice puede ver que en la solicitud hay algunos atributos que se exige estén verificados por un *Issuer* determinado y bajo un *Credential Definition* concreto. Estos atributos son "Grado", "Estado", "Número de la Seguridad Social" y "Nota media". Además, se solicita que estén verificados bajo el *Credential Definition* identificado como *Faber Transcript* (ya que la empresa confía en la universidad Faber). Analizando con un poco más de detalle vemos que los valores de los atributos "Grado", "Estado" y "Número de la seguridad Social" deben ser revelados, pero que para "Nota media" es suficiente con un *Zero Knowledge Proof* (ZKP), es decir, que solo se debe confirmar si la nota media está por encima de 4 (con un sí o un no). El resto de atributos ("Nombre", "Apellidos" y "Número de teléfono") pueden ser auto-afirmados (*self-asserted*).

Recibida y analizada la petición de evidencias, Alice comprueba en su *wallet* de qué credenciales dispone que cumplan con lo solicitado. Efectivamente dispone de la evidencia *Faber Transcript Credential*, emitida por Faber bajo el esquema *Transcript* (v1.2). Con estas evidencias Alice prepara la prueba que solicita ACME y se la hace llegar, haciendo referencia al *Faber Transcript* para atestiguar los atributos "Grado", "Estado", "Número de la Seguridad Social" y "Nota media" (cómo ZKP) e

incluyendo los atributos auto-afirmados “Nombre”, “Apellidos” y “Número de teléfono”.

Cuando ACME recibe la prueba enviada y autenticada por Alice, la analiza y ve que para aquellos atributos para los que había solicitado verificación se hace referencia a un *Credential Definition* llamado *Faber Transcript*. ACME obtiene la información del *Credential Definition* y verifica que efectivamente los atributos que se referencian en la prueba de Alice son atributos certificados por Faber. Con esto, y gracias a la confianza y a la reputación de Faber, en ACME pueden estar seguros de que Alice cumple con los requisitos para la vacante laboral.

4.2.5 OBTENIENDO EL CERTIFICADO LABORAL

Una vez obtenido el empleo, Alice pasa a ser empleada de ACME. Siguiendo con la política definida a raíz del proyecto piloto, ACME generará para los trabajadores que lo soliciten, una evidencia en la que se atestigua que son empleados de la empresa. ACME utiliza para ello el *schema* definido y publicado por el Ministerio de Empleo llamado *Job-Certificate* (v0.2).

Después de 10 meses trabajando, Alice solicita la evidencia conforme es empleada a ACME, incluyendo una referencia a su *master secret* en la solicitud. ACME que ya ha instanciado el esquema *Job-Certificate* (v0.2) y ha creado un *Credential Definition* llamado *ACME Job Certificate*, procesa la solicitud de Alice y procede a generarle las credenciales.

Siguiendo el esquema *Job-Certificate* (v0.2) ACME incluye en las credenciales los atributos “Nombre”, “Apellidos”, “Salario” (bruto anual), “Estado del empleado” (contrato temporal o fijo) y “Experiencia” (en meses), con los valores que le corresponden a Alice. Cuando ella recibe las credenciales las almacena en su *wallet*, sabiendo que las necesitará para poder pedir el crédito personal que necesita para comprarse un coche.

4.2.6 APLICANDO PARA EL CRÉDITO PERSONAL

Después de pasar por una de las oficinas de Thrift Bank, Alice ya tiene claras las condiciones del producto financiero que necesita para comprarse el coche que necesita. En la oficina también han informado a Alice de que todos los trámites administrativos para formalizar la solicitud del crédito personal y para el proceso de KYC (*Know Your Customer*) se realizará de manera telemática, a través de la *Sovrin Network*. Alice, que ya está familiarizada con este ecosistema no ve ningún problema en ello y al llegar a casa establece una conexión con Thrift Bank, obteniendo la solicitud de evidencias para la petición de un crédito personal (*Loan Application Basic Proof Request*).

Alice analiza qué atributos de su identidad debe evidenciar para poder solicitar el crédito personal. En la solicitud se piden como atributos verificables relacionados con el puesto de trabajo “Estado del empleado” (contrato temporal o fijo), “Salario” (bruto mensual) y “Experiencia” (en meses). El valor del primer atributo debe ser revelado, pero para los otros dos (salario y experiencia) es suficiente con una *Zero Knowledge Proof*.

Para el salario Alice debe confirmar que está por encima de los 2.400 euros brutos mensuales, pero no es necesario que revele el importe total de sus ingresos. Para la experiencia debe confirmar que tiene más de 1 mes de antigüedad en la empresa, pero no es necesario que revele cuánto tiempo lleva trabajando en ella. La solicitud de evidencias de Thrift Bank también especifica que los atributos verificables deben demostrarse con el *Credential Definition* llamado *ACME Job Certificate* (ya que el banco confía en la empresa ACME).

Siendo así Alice comprueba en su *wallet* que efectivamente dispone de una instancia de este *Credential Definition*. Con ello Alice prepara la respuesta a la *Proof Request* de Thrift Bank y se la hace llegar, haciendo referencia a su *master secret* para dejar constatación de que ese conjunto de evidencias se ha emitido (por ella misma) para ella y no para cualquier otro.

Como vemos, Alice solo ha revelado que es una trabajadora con contrato de trabajo permanente (“Estado del empleado”), que su salario está por encima de los 2.400 euros brutos mensuales y que tiene más de 1 mes de antigüedad en su actual empleo. Con esto es suficiente para que Thrift Bank decida, en base a su política de gestión de riesgos, si puede conceder el préstamo al solicitante (notemos que para tomar esta decisión el banco no ha necesitado saber ni el nombre de Alice).

Después de recibir la respuesta de Alice, Thrift Bank comprueba que efectivamente los atributos verificables están atestiguados por ACME, y que todos ellos satisfacen las condiciones necesarias para obtener un crédito personal. No obstante, la legislación obliga a los bancos a verificar la identidad de todos sus clientes para, por ejemplo, luchar contra el blanqueo de capitales. Cumpliendo con la regulación, Thrift Bank necesita completar su proceso de KYC para poder formalizar el crédito con Alice. Este proceso básicamente consiste en obtener aquellos datos personales que permiten al estado identificar a los clientes. Concretamente “Nombre”, “Apellidos” y “Número de la Seguridad Social”.

Para seguir con el proceso Thrift Bank hace llegar a Alice una segunda *Proof Request* llamada *Loan Application KYC*, en la que se pide se evidencien el nombre, los apellidos y el número de la Seguridad Social, sin especificar con qué *Credential Definition*. Al recibir la petición Alice comprueba su *wallet* y ve que hay atributos que puede demostrar con los *Credentials* que posee: el *Faber Transcript* y el *ACME Job Certificate*. El nombre y los apellidos los podrá demostrar con cualquiera de los dos, y el número de la seguridad social con el que en su día le emitió Faber.

Con esto Alice prepara la respuesta a la *Proof Request* de Thrinf Bank, y se la hace llegar junto con su *master secret* para, de nuevo, dejar constatación de que ese conjunto de evidencias se ha emitido (por ella misma) para ella y no para cualquier otro.

Al haber indicado Alice con qué instancias de los *Credential Definition* pueden verificarse los atributos, Thrift Bank no tiene ningún problema a la hora de validarlos, dando por concluido el proceso de KYC y concediendo el crédito personal a Alice.

No obstante, al tratarse de datos PII (*Personally Identifiable Information*), Thrift Bank deberá gestionarlos acorde a la normativa vigente (GDPR), aunque los haya obtenido a través de un sistema como

Sovrin. De lo que podemos estar seguros gracias a Sovrin es de qué Alice y Thrift Bank han compartido únicamente los datos estrictamente necesarios y de qué todo se ha hecho por medios digitales, sin necesidad de ninguna tercera parte certifique la veracidad de los datos intercambiados.

5. ANÁLISIS TÉCNICO DEL CASO DE USO EN EL ENTORNO PRÁCTICO

Descrito y comprendido el Caso de Uso, en este apartado se analizará técnicamente su implementación, en base al código de la librería Libindy (Artemkaaas & varios, 2018). Se pretende con este análisis afianzar la comprensión del funcionamiento de la *Sovrin Network*, relatando cómo se comporta cada uno de sus actores desde el punto de vista técnico y qué objetos participan en cada acción. Para ello se seguirán los acontecimientos relatados en el Caso de Uso identificando cada intercambio de información entre actores con un diagrama de secuencia. Cuando sea relevante se identificarán también las funciones de la librería llamadas.

5.1 DANDO DE ALTA A LOS ACTORES EN LA SOVRIN NETWORK

5.1.1 GETTING TRUST ANCHOR CREDENTIALS – ONBOARDING

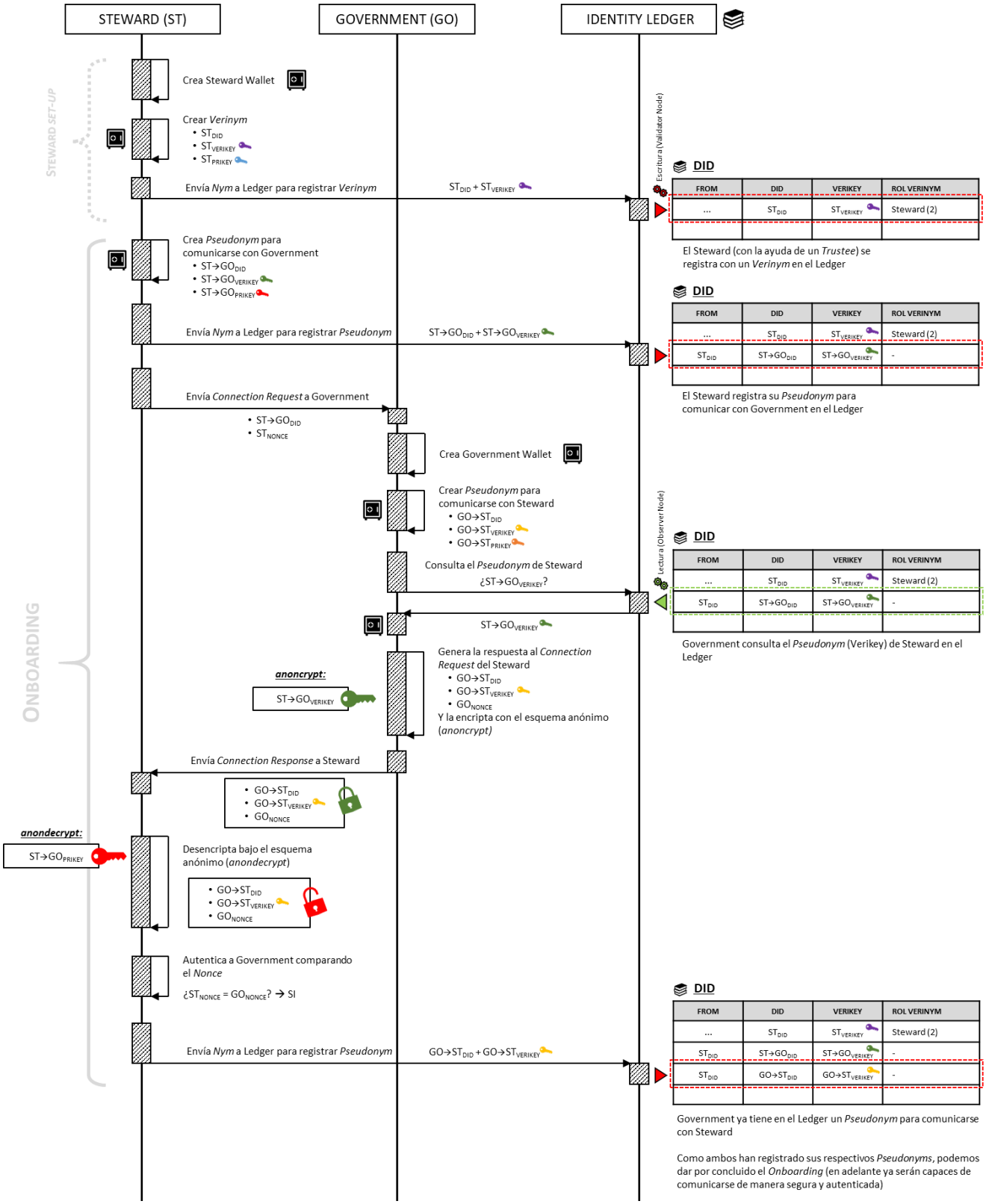

Ilustración 8. Onboarding genérico para los Actores

5.1.2 GETTING TRUST ANCHOR CREDENTIALS – GETTING VERINYM

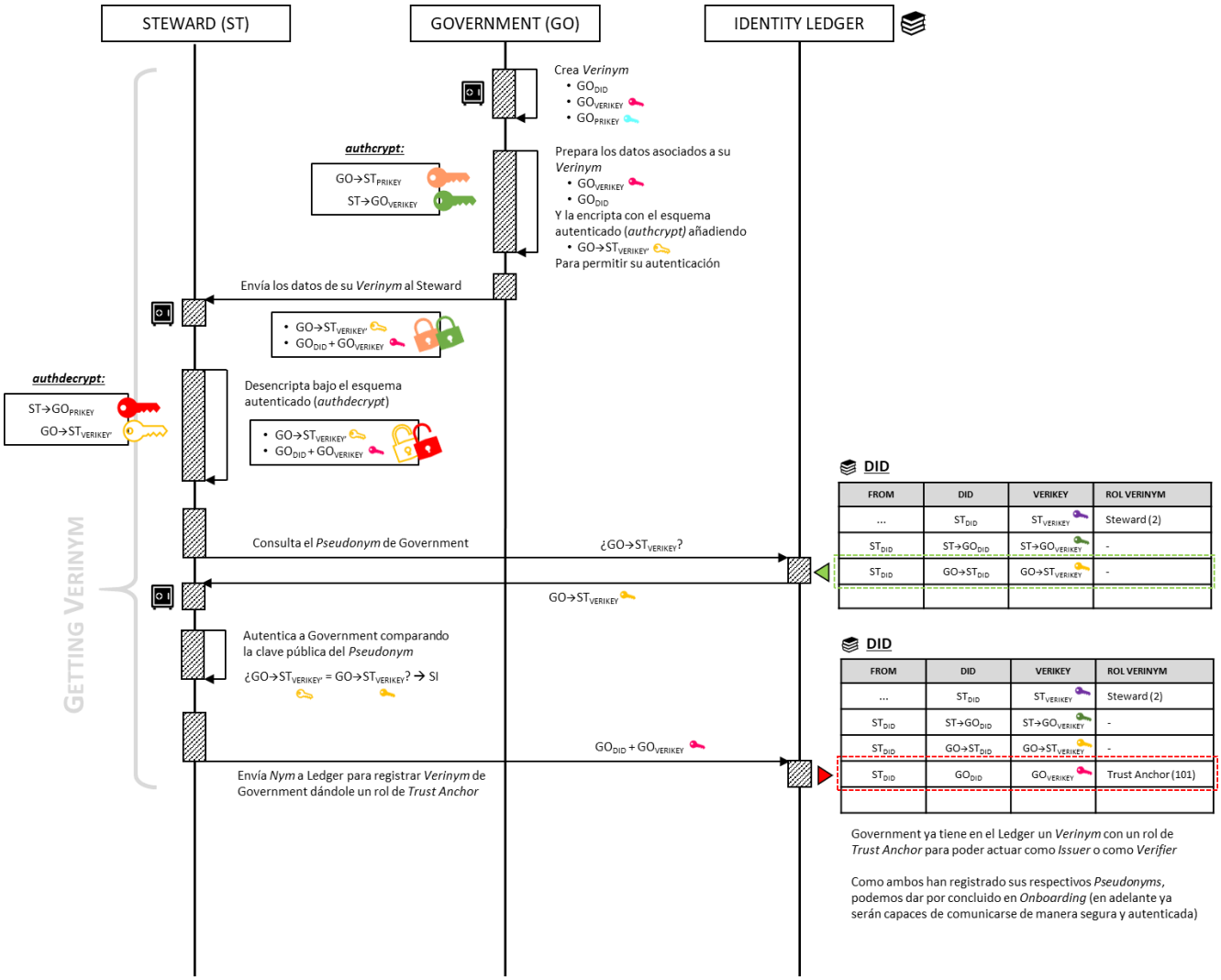

Ilustración 9. Getting Verinym genérico para los Actores

El proceso de *Onboarding* (5.1.1 Getting Trust Anchor credentials – Onboarding) y de *Getting Verinym* (5.1.2 Getting Trust Anchor credentials – Getting Verinym) se repetirá entre el Steward y cada uno de los actores que deban asumir un papel de *Issuer* o *Verifier*, ya que es necesario que todos ellos tengan un rol de *Thrust Anchor* asignado para poder escribir en el Ledger. En nuestro escenario estos actores serán:

- Government (actuará como *Issuer*)
- Faber (actuará como *Issuer*)
- Acme (actuará como *Issuer* y como *Verifier*)
- Thrift Bank (actuará como *Verifier*)

5.2 ESTANDARIZANDO EL CONTENIDO DE LAS EVIDENCIAS

5.2.1 FABER CREDENTIAL DEFINITION SETUP

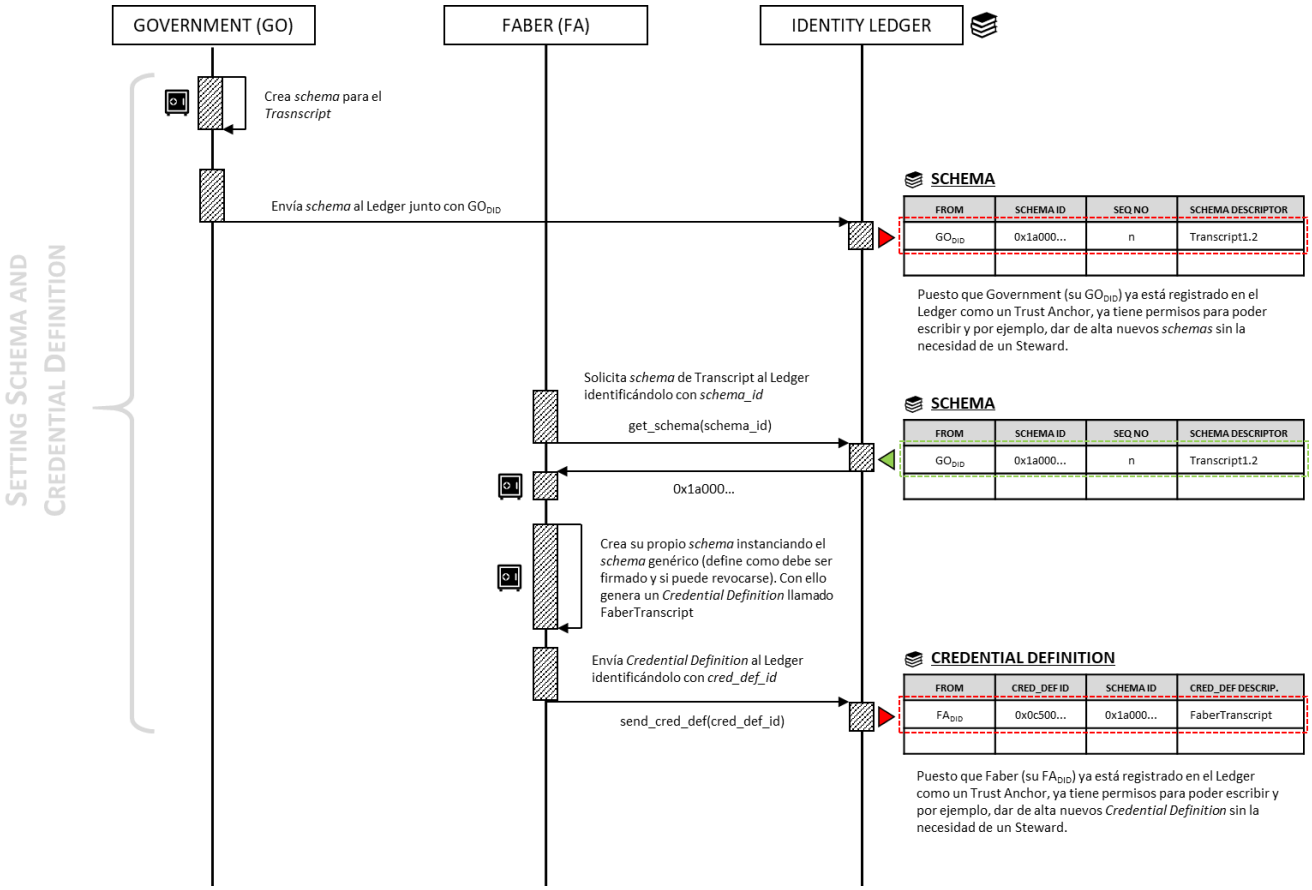

Ilustración 10. Credential Definition Setup de Faber

5.2.2 ACME CREDENTIAL DEFINITION SETUP

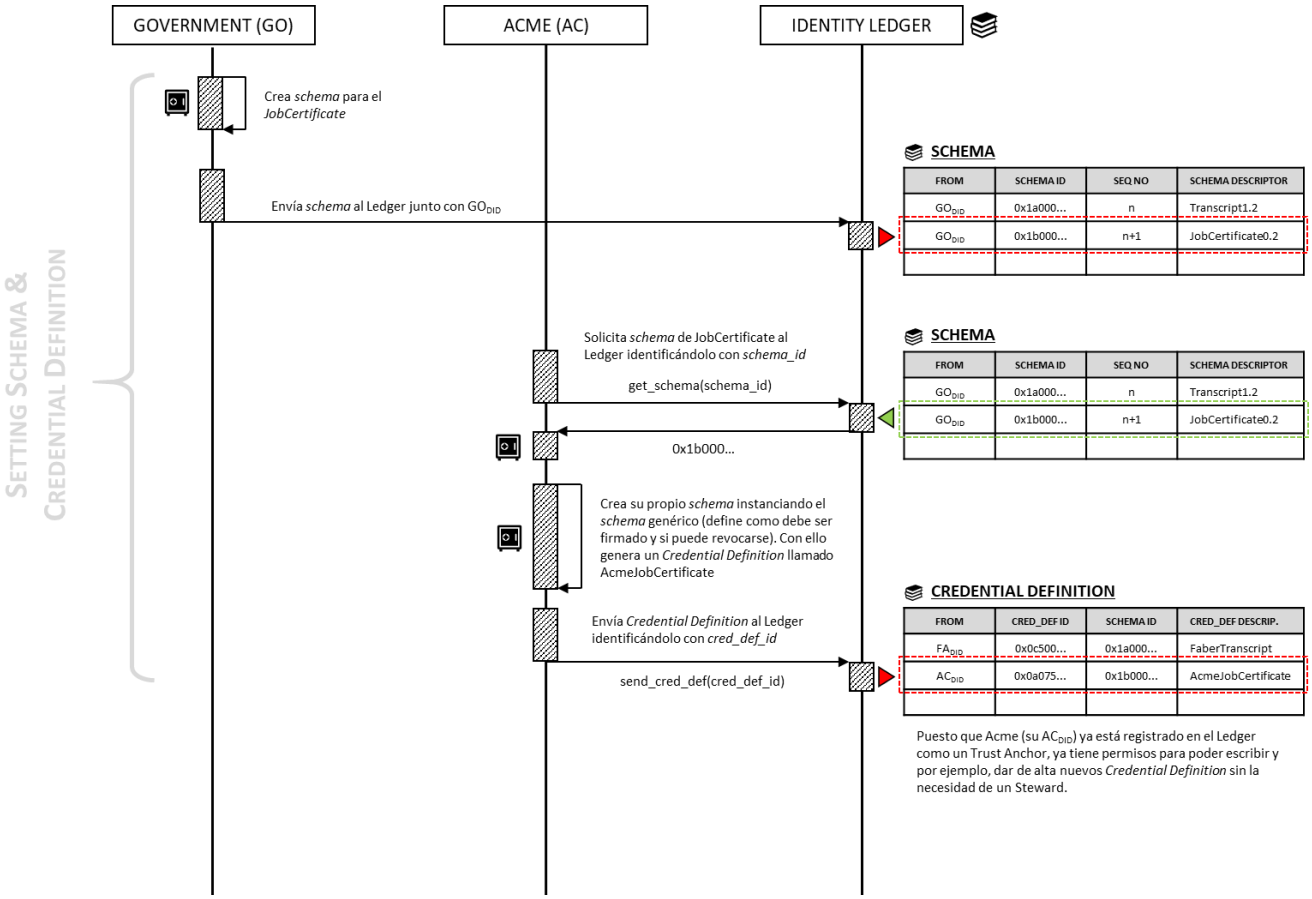

Ilustración 11. Credential Definition Setup de Acme

5.3 OBTENIENDO EL CERTIFICADO DE ESTUDIOS

5.3.1 GETTING TRANSCRIPT WITH FABER – ONBOARDING

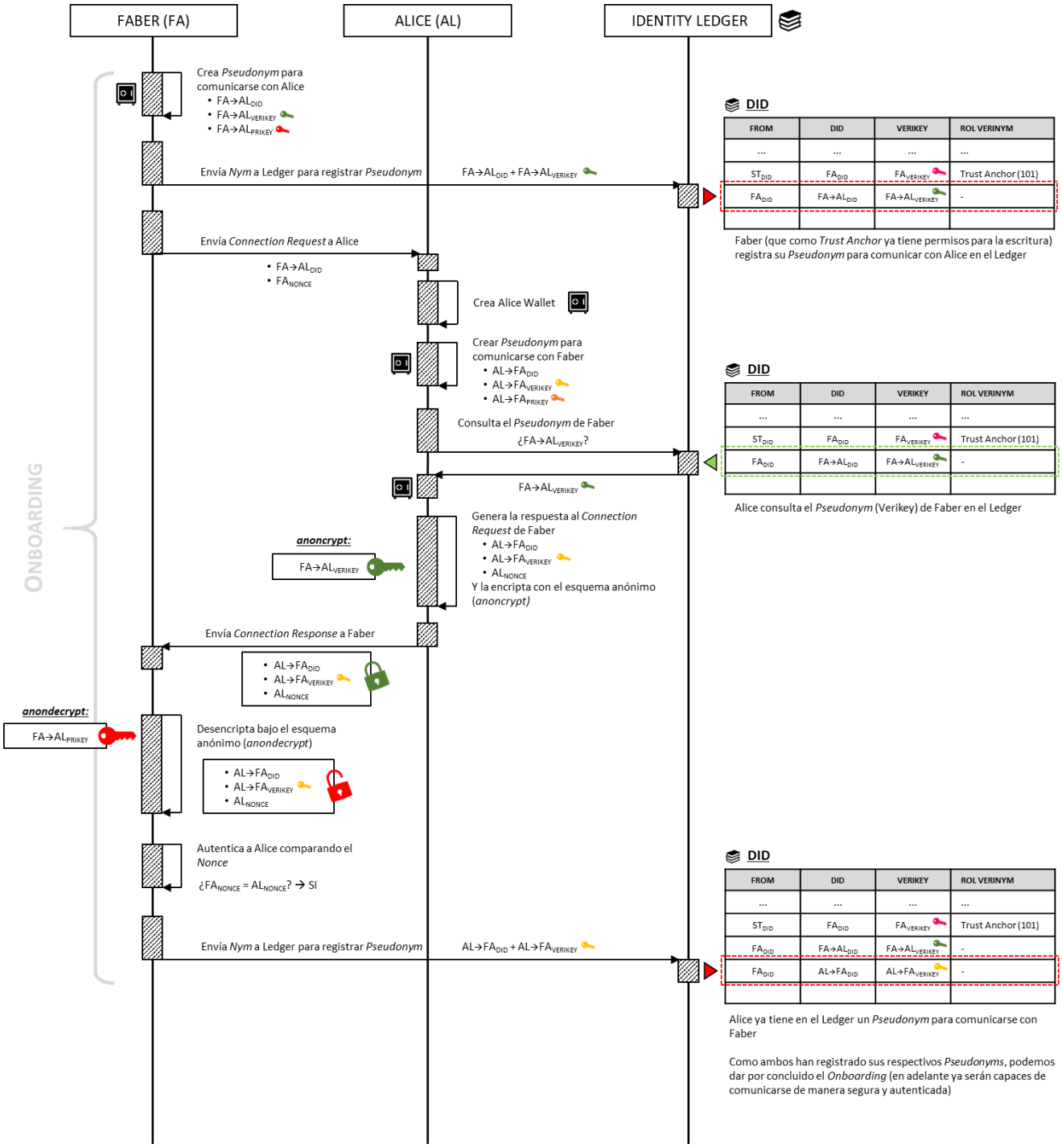

Ilustración 12. Onboarding entre Faber y Alice

5.3.2 GETTING TRANSCRIPT WITH FABER – GETTING TRANSCRIPT CREDENTIAL

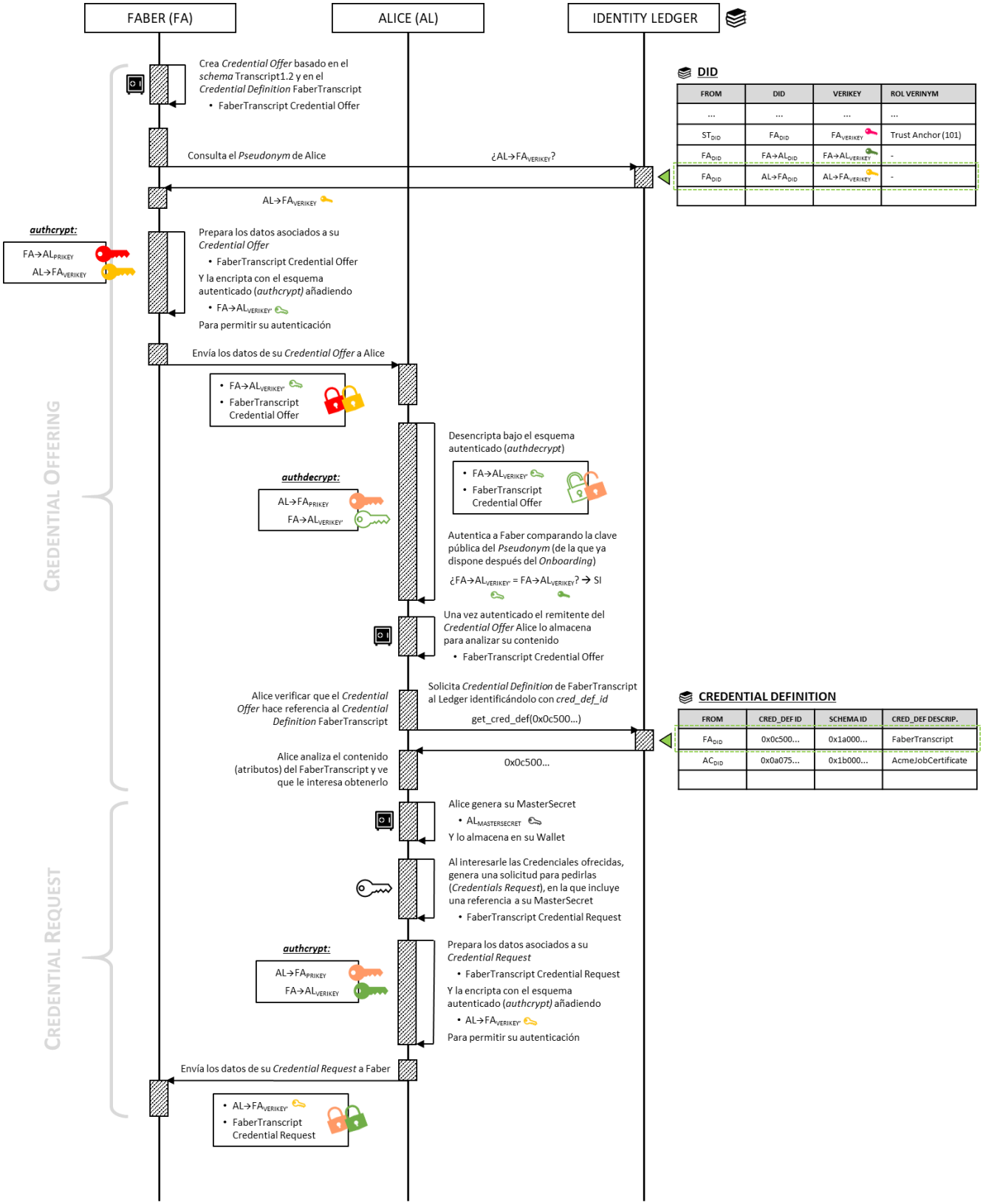

Ilustración 13. Credenciales (Transcript) para Alice [1/2]

(continua 5.3.2 Getting Transcript with Faber – Getting Transcript Credential)

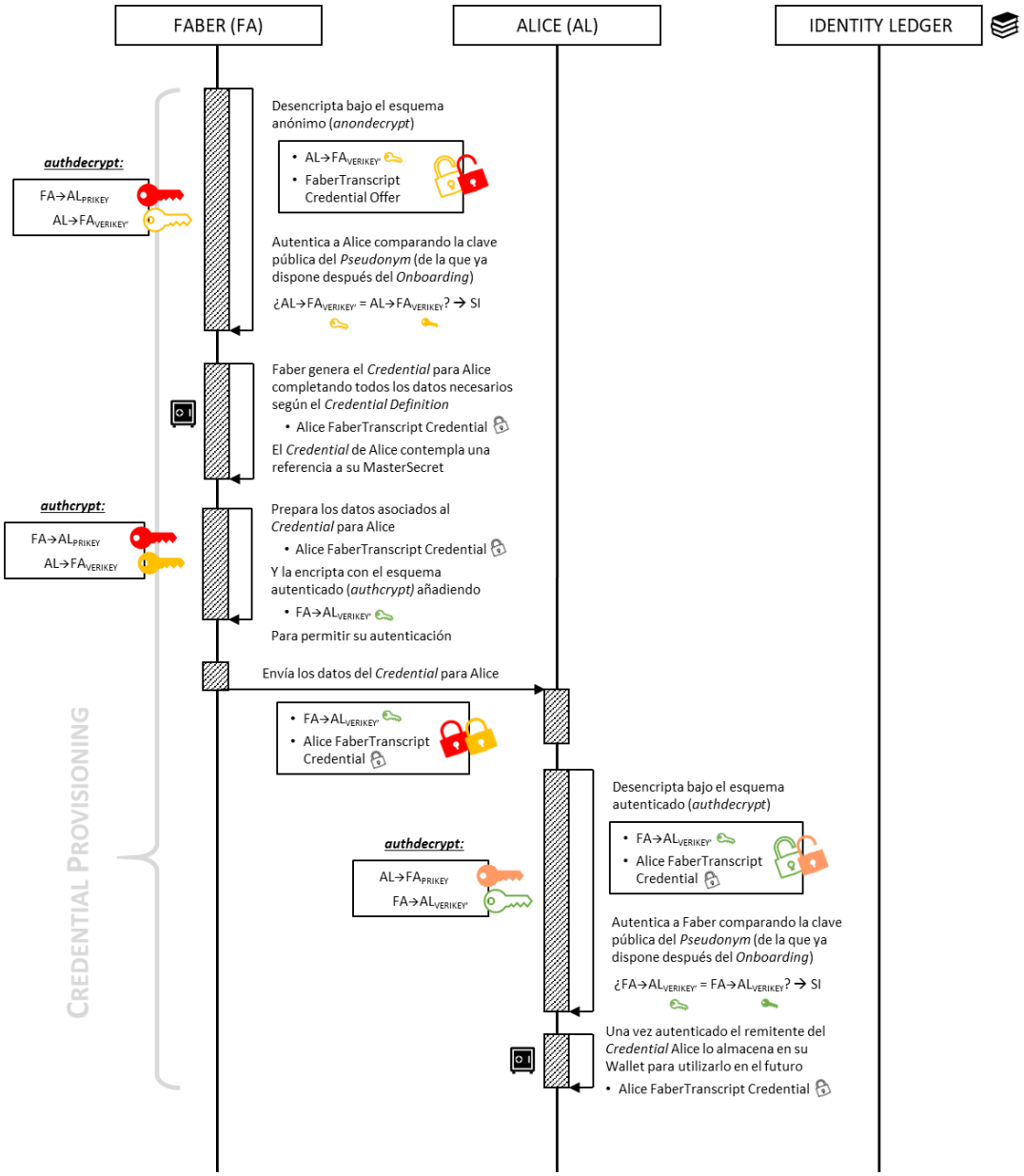

Ilustración 14. Credenciales (Transcript) para Alice [2/2]

5.4 APLICANDO A LA VACANTE LABORAL

5.4.1 APPLY FOR THE JOB WITH ACME – ONBOARDING

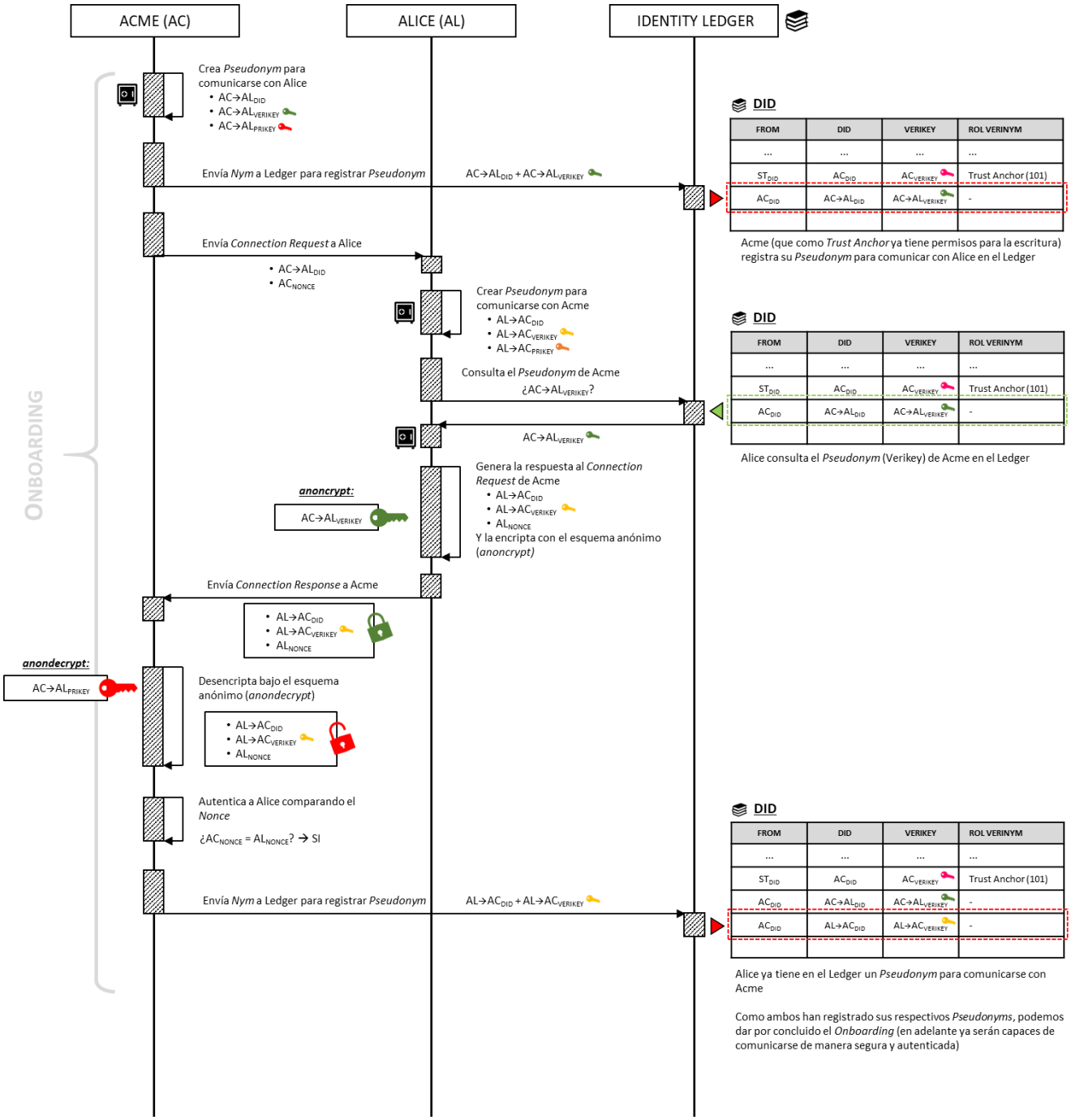

Ilustración 15. Onboarding entre Acme y Alice

5.4.2 APPLY FOR THE JOB WITH ACME – TRANSCRIPT PROVING

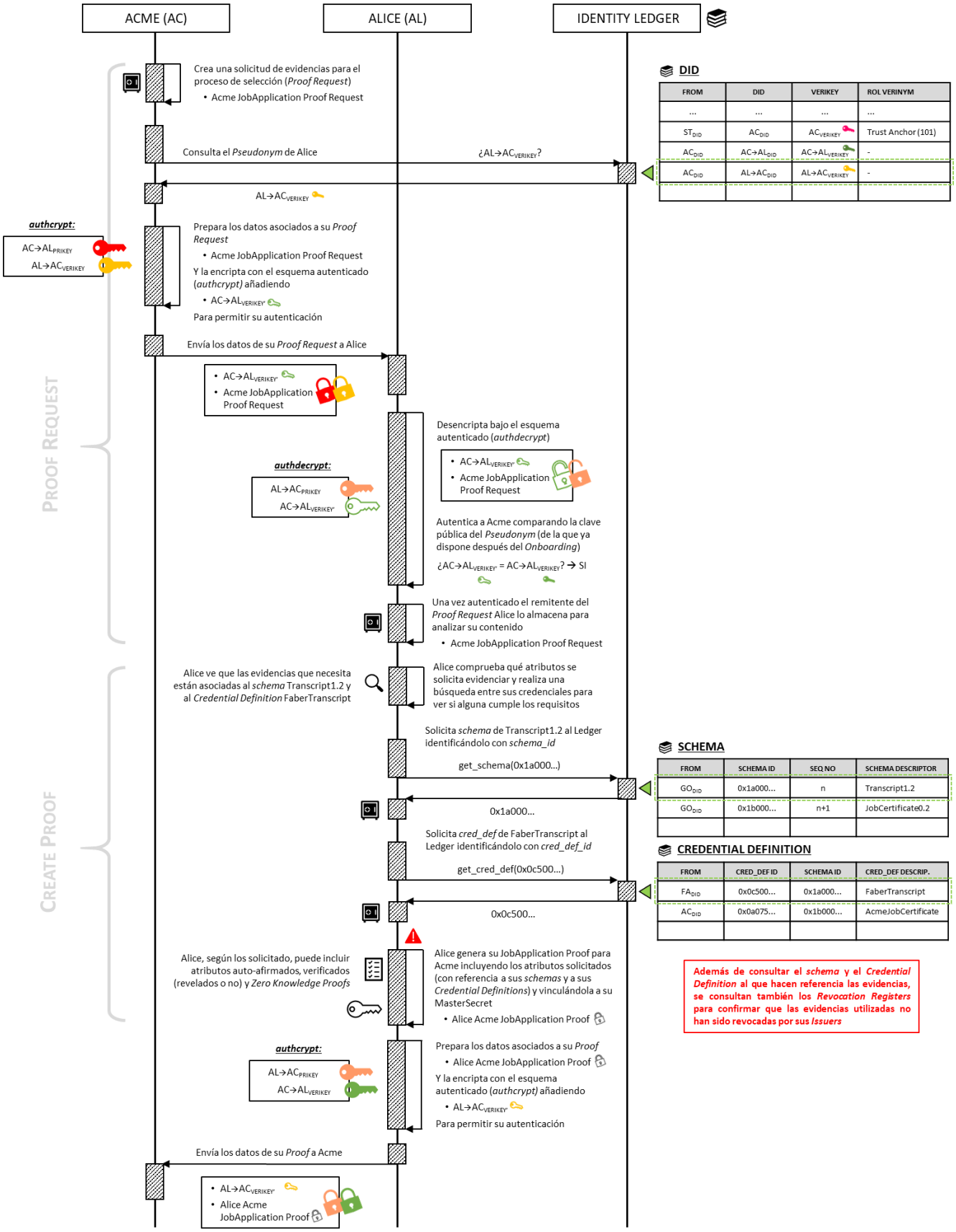

Ilustración 16. Evidencias (JobApplication Proof) por parte de Alice [1/2]

(continua 5.4.2 Apply for the job with Acme – Transcript proving)

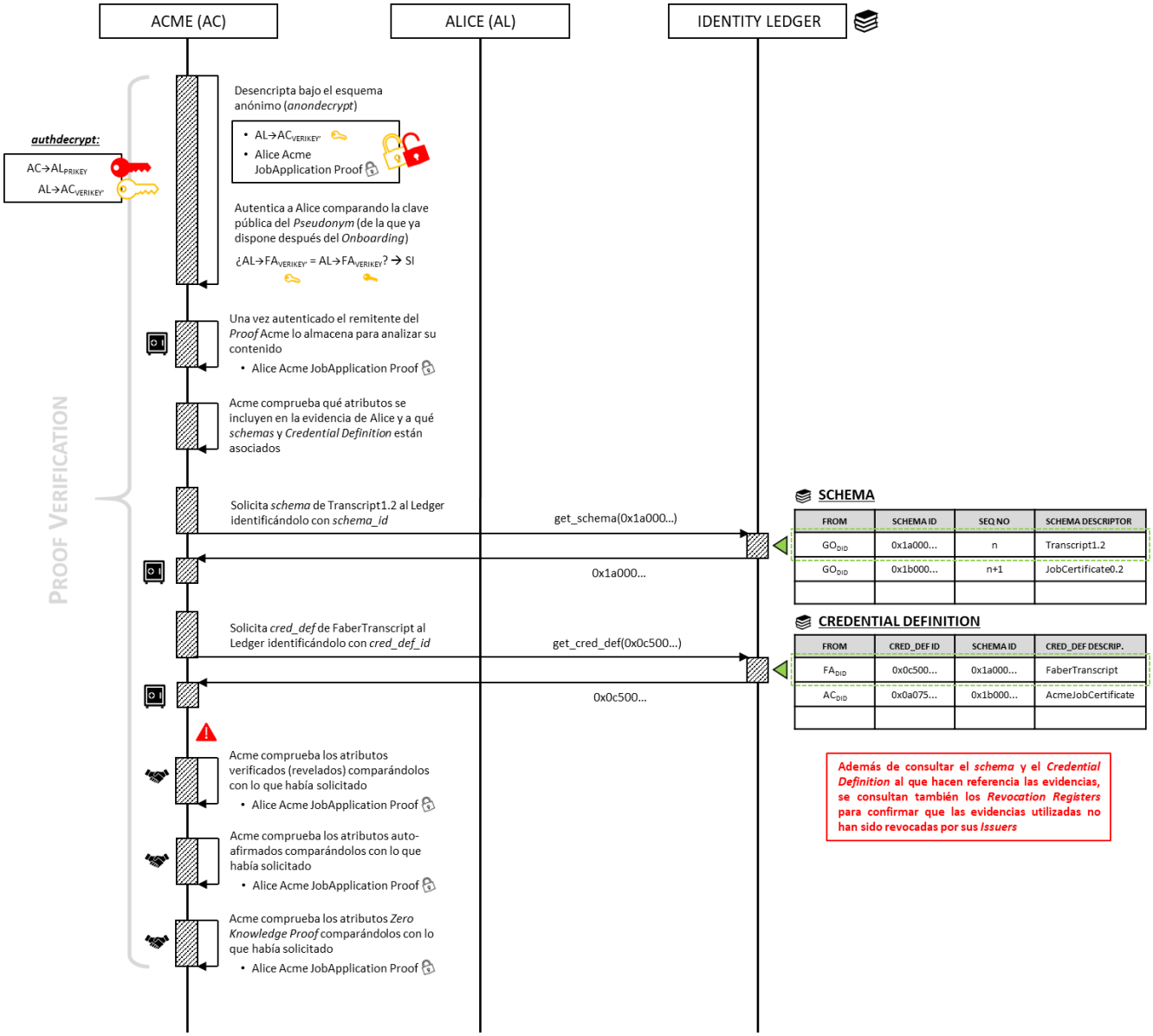

Ilustración 17. Evidencias (JobApplication Proof) por parte de Alice [2/2]

5.5 OBTENIENDO EL CERTIFICADO LABORAL

5.5.1 APPLY FOR THE JOB WITH ACME – GETTING JOB-CERTIFICATE CREDENTIAL

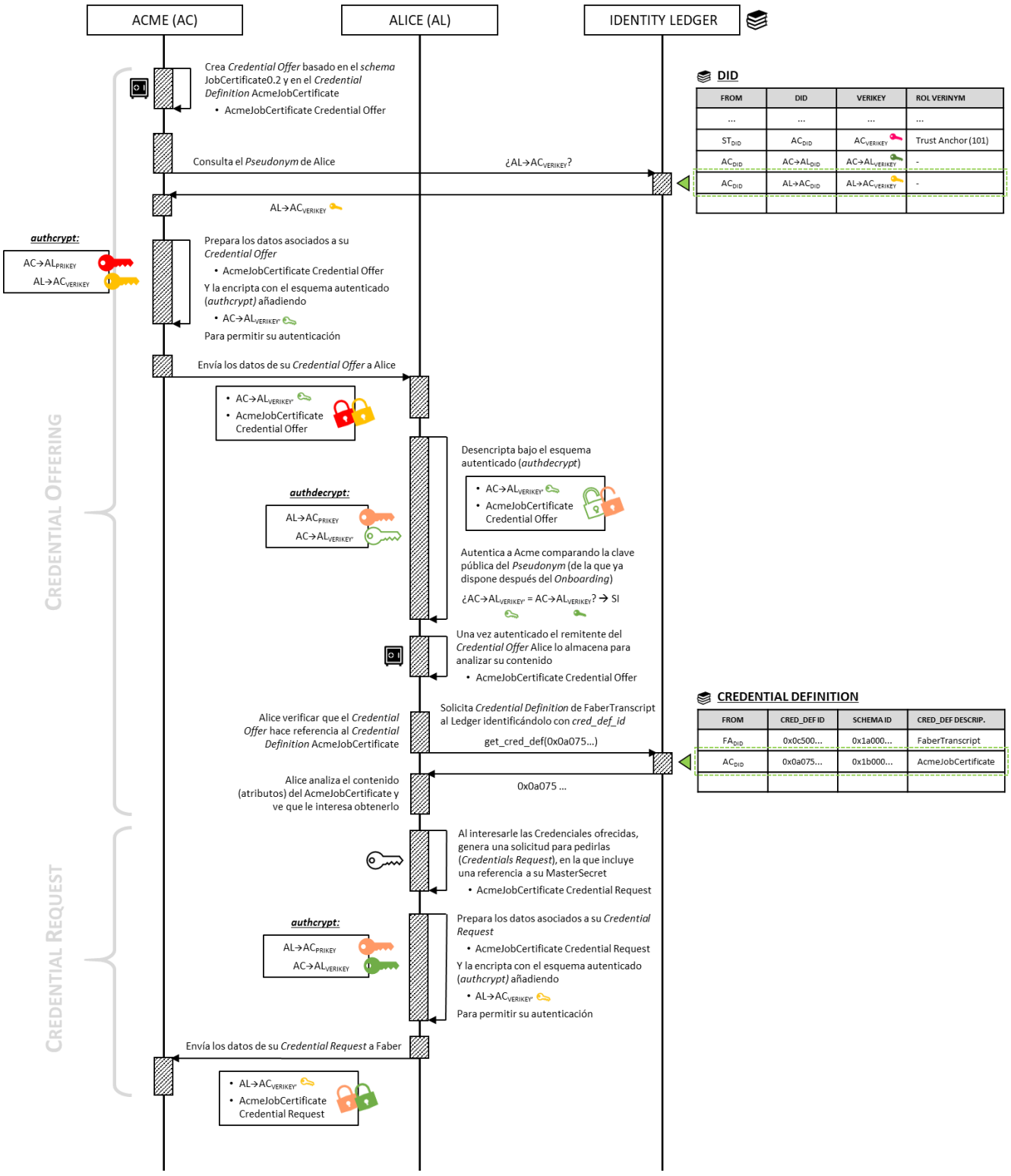

Ilustración 18. Credenciales (JobCertificate) para Alice [1/2]

(continua 5.5.1 Apply for the job with Acme – Getting Job-Certificate Credential)

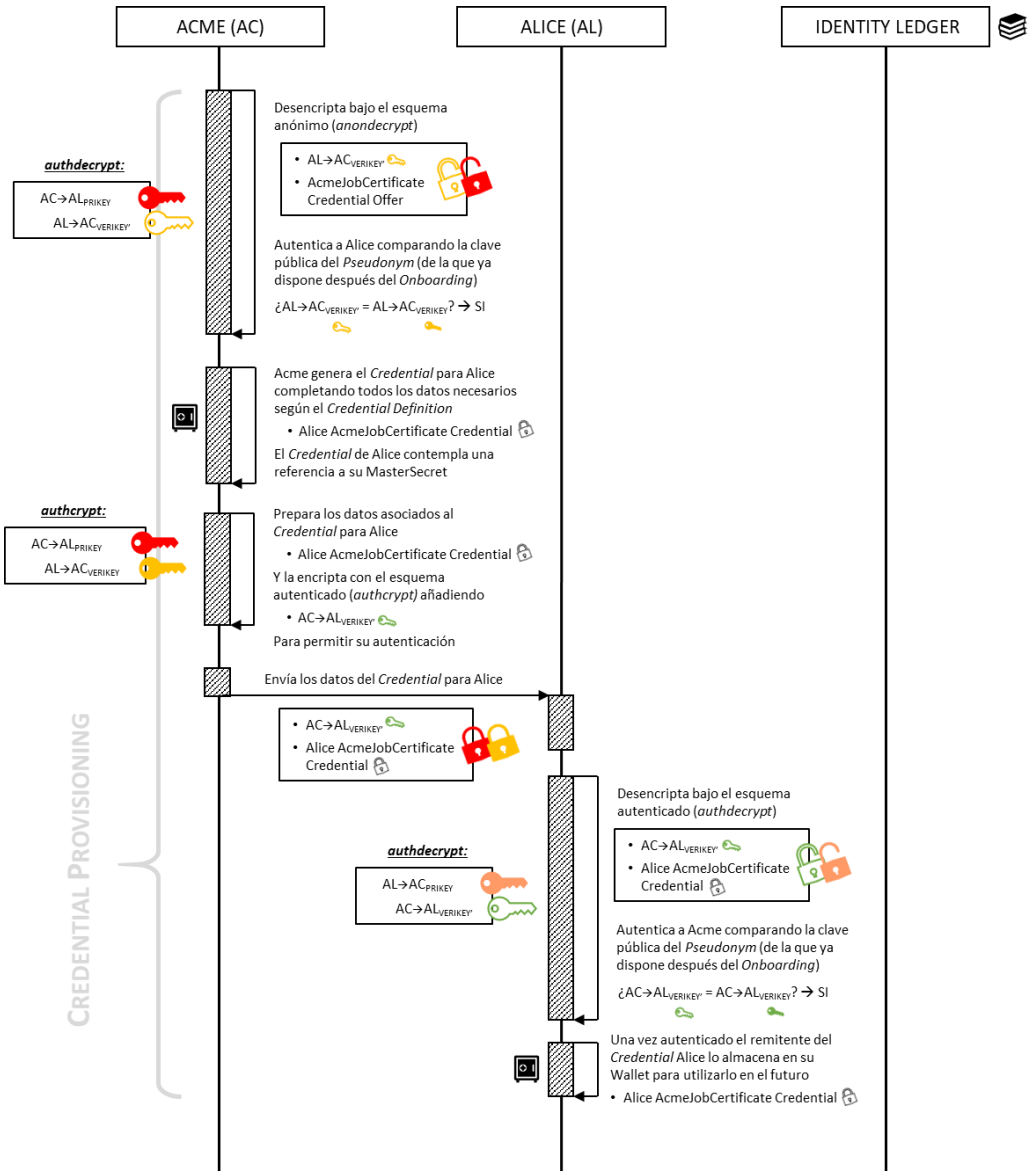

Ilustración 19. Credenciales (JobCertificate) para Alice [2/2]

5.6 APLICANDO PARA EL CRÉDITO PERSONAL

5.6.1 APPLY FOR THE LOAN WITH THRIFT – ONBOARDING

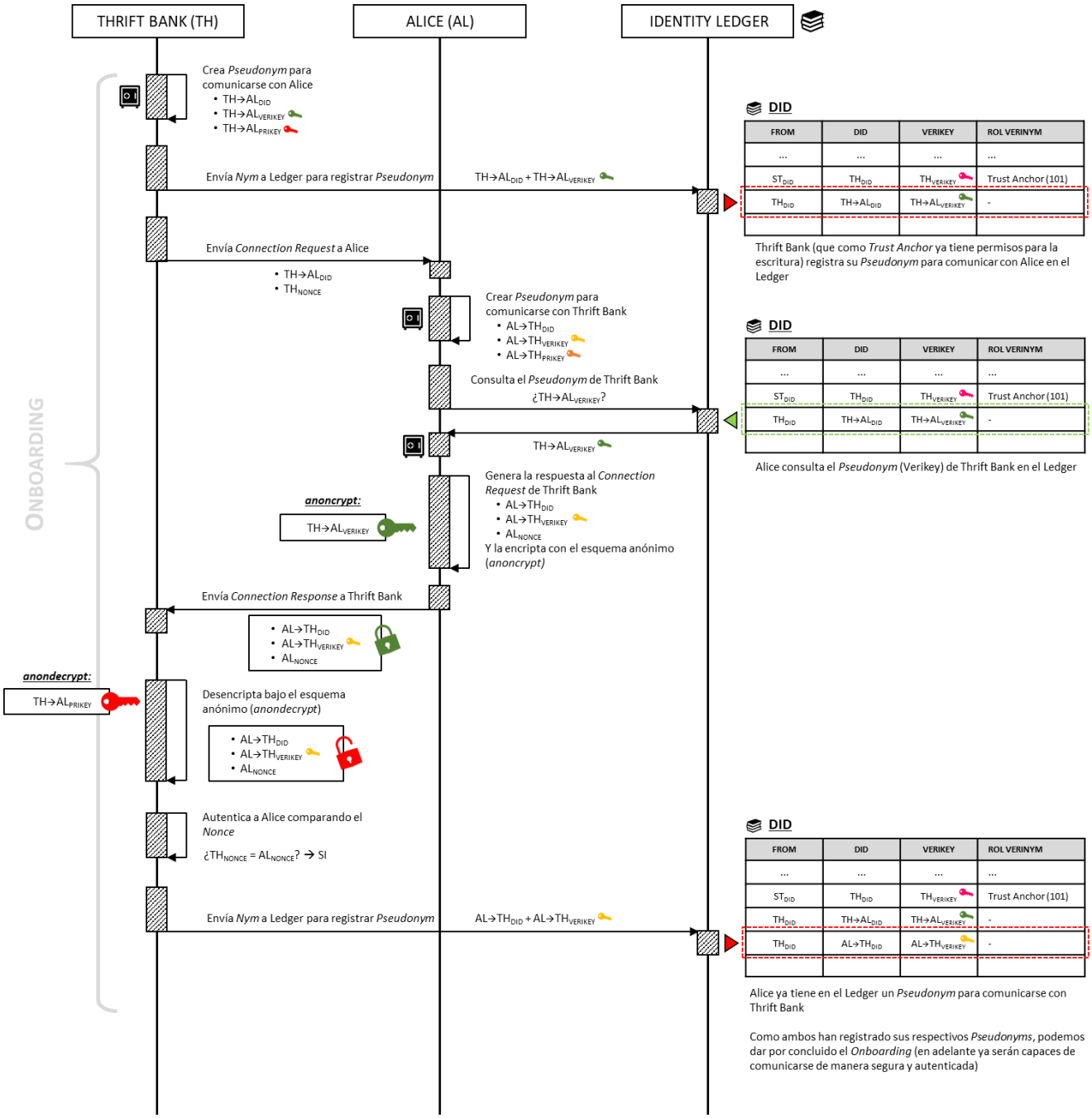

Ilustración 20. Onboarding entre Thrift Bank y Alice

5.6.2 APPLY FOR THE LOAN WITH THRIFT – JOB-CERTIFICATE PROVING

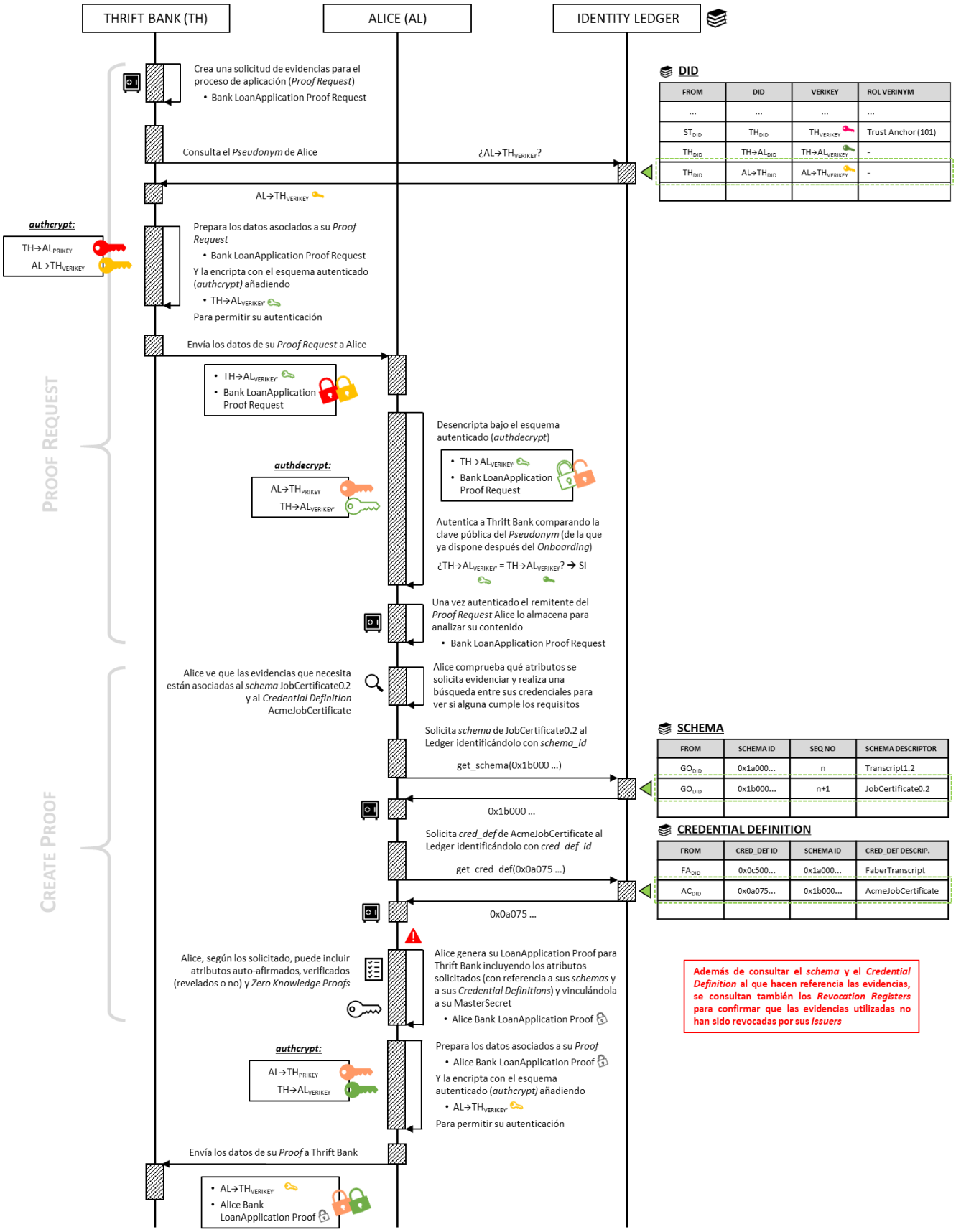

Ilustración 21. Evidencias (LoanApplication Proof) por parte de Alice [1/2]

(continua 5.6.2 Apply for the loan with Thrift – Job-Certificate proving)

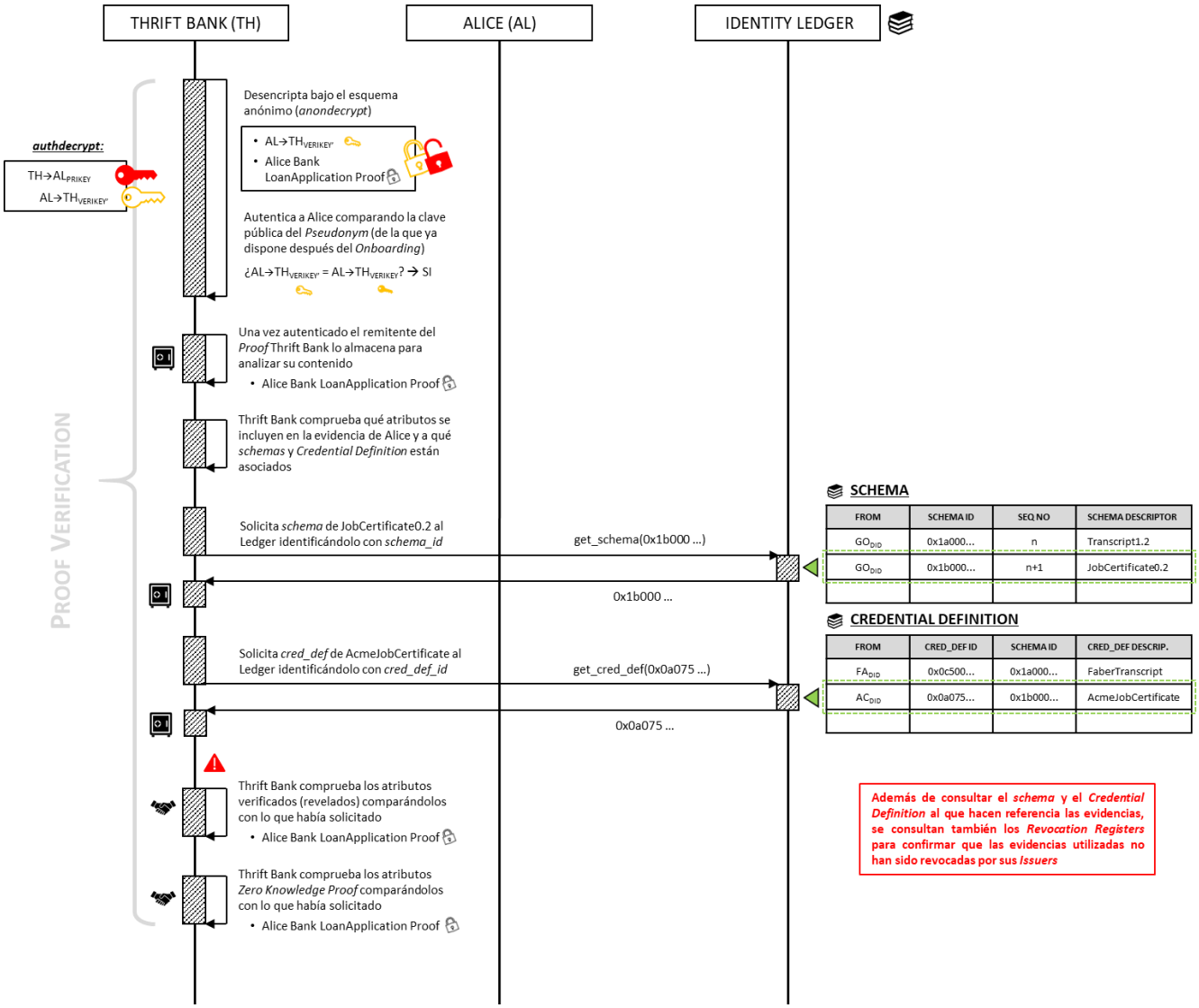

Ilustración 22. Evidencias (LoanApplication Proof) por parte de Alice [2/2]

5.6.3 APPLY FOR THE LOAN WITH THRIFT – TRANSCRIPT AND JOB-CERT. PROVING

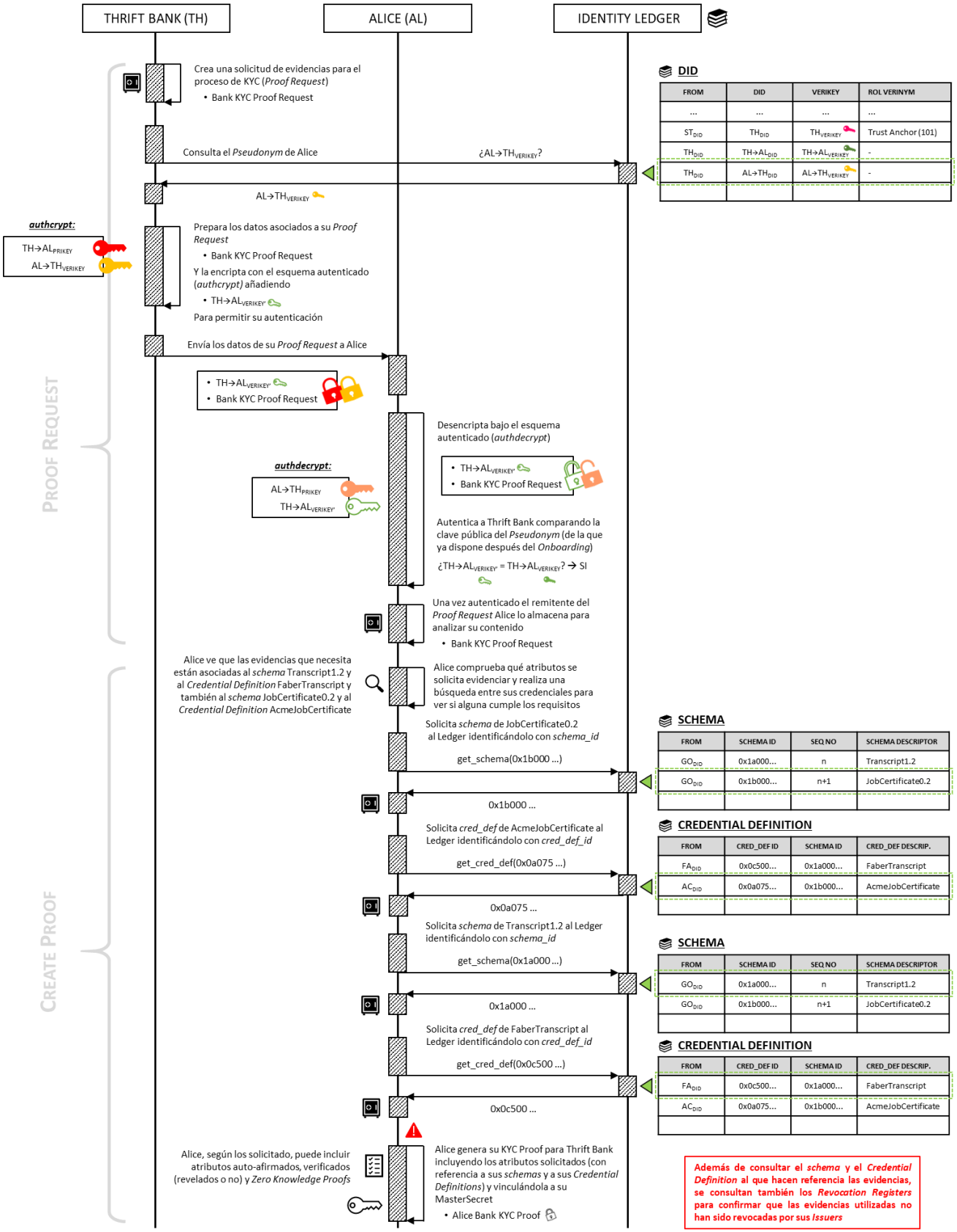

Ilustración 23. Evidencias (KYC Proof) por parte de Alice [1/2]

(continua 5.6.3 Apply for the loan with Thrift – Transcript and Job-Cert. proving)

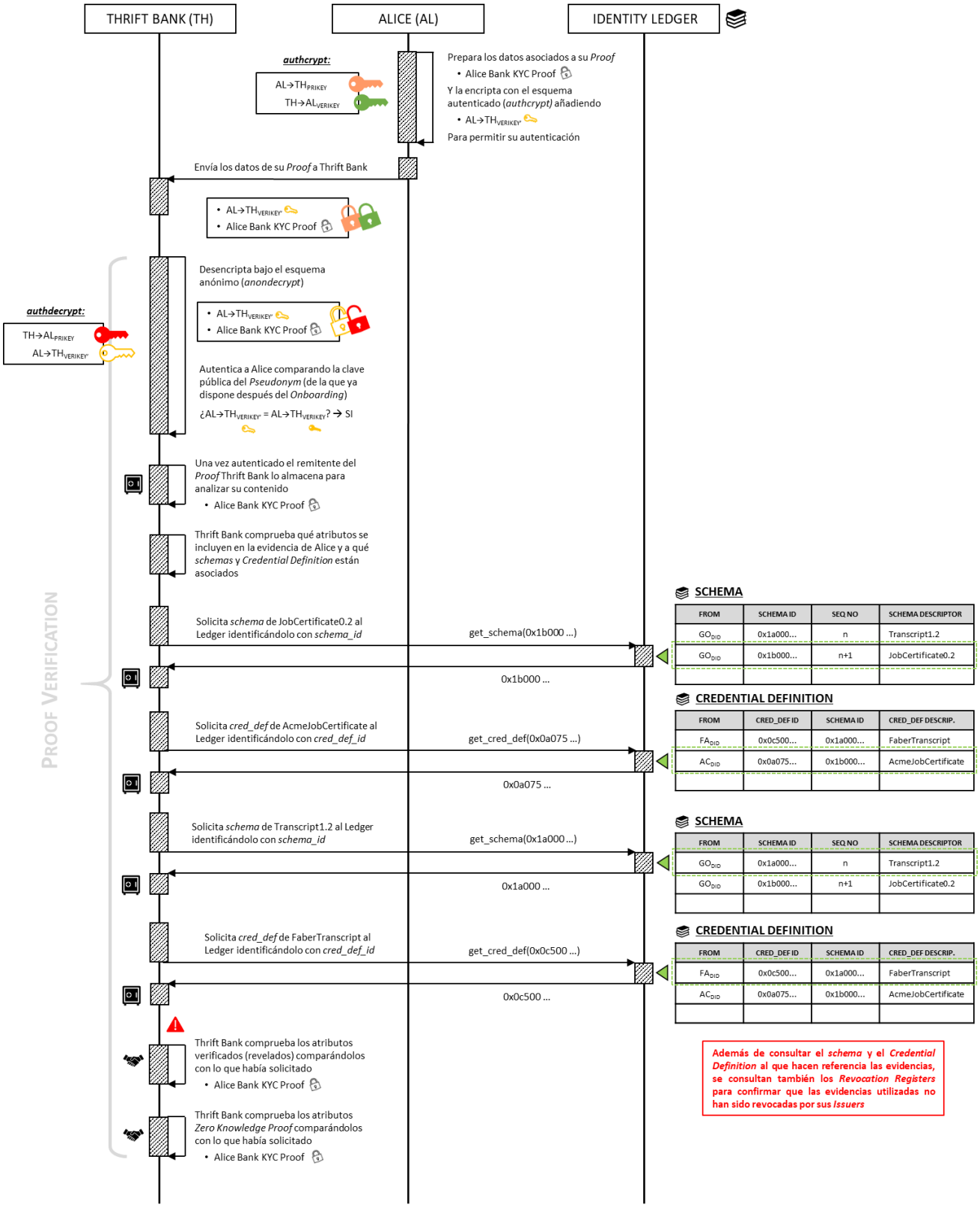

Ilustración 24. Evidencias (KYC Proof) por parte de Alice [2/2]

6. CONCLUSIONES

Después de entrar en contacto con el ecosistema de Sovrin y ver cómo este plantea la implementación de un paradigma de gestión soberana de la identidad, podemos concluir varias cosas.

1. La gestión soberana de la identidad es el mecanismo hacia el cual el mercado debe dirigirse, puesto que es el más justo para los usuarios y la técnica le da viabilidad técnica suficiente. Vemos también que es un paradigma por el cual las grandes entidades ya se están interesando, siendo ejemplo de ello el *World Economic Forum* o las empresas privadas que constituyen la *Sovrin Foundation* (https://sovrin.org/stewards/).
2. También es posible prever resistencia por parte de las corporaciones digitales que hoy obtienen crasos beneficios fundamentándose en el modelo de gestión actual. Estas empresas se valen de la falta de conciencia de la ciudadanía y de la falta de regulación para apropiarse de la identidad digital de los usuarios. Además, actualmente se ofrecen cómo proveedores principales de identidad (IdP), generando la sensación de estar ofreciendo un servicio gratuito cuando realmente obtienen los beneficios comerciando con nuestros datos. Será necesario vencer esta resistencia y ofrecer vías de comercialización alternativas a estos actores para que el nuevo paradigma de gestión de la identidad pueda establecerse. Deberán ser los gobiernos y los ciudadanos los que obliguen al mercado a trabajar bajo este nuevo paradigma ya que, de lo contrario, este no se impondrá como estándar *de facto*.
3. En cuanto a la propuesta de Sovrin, se trata de un proyecto abierto que se encuentra en fase de incubación, por lo que todavía tiene un gran camino por delante. Hoy en día no existe un nivel de desarrollo suficiente como para poder utilizar sus herramientas, ni tan siquiera en un entorno de pruebas piloto. No obstante, parece que la comunidad está trabajando en ello, por lo que es posible que, en los próximos años, el entorno esté lo suficiente maduro como para plantear iniciativas a pequeña y mediana escala. Si finalmente Sovrin se impone como estándar, no será hasta dentro de 5 o 10 años que su uso esté asentado y aceptado por parte de la mayoría de los usuarios. Podemos recordar el caso del Bitcoin, concebido en 2009 y a fecha de hoy, todavía con muchas reticencias en cuanto a su uso cotidiano. La falta de concienciación y de conocimientos por parte de los usuarios y el miedo a perder el *statu quo* de las grandes corporaciones, son las principales barreras de estas tecnologías disruptivas.
4. La propuesta de la *Sovrin Foundation* parece que cubre las necesidades que podría tener un sistema de gestión soberana de la identidad. Canaliza tanto las necesidades tecnológicas como las necesidades organizativas, descentralizando la gestión y la operación de la infraestructura. Parece que hay consenso en la

Comunidad en lo relativo a Sovrin como herramienta fundamental, ya que no se detectan propuestas alternativas que estén a su mismo nivel en cuanto a difusión y adopción. No obstante, habrá que ver como evoluciona toda la comunidad Blockchain y cómo los gobiernos regulan al respecto, para dar a estos sistemas el respaldo jurídico que necesitan como garantes de la integridad de la información.

En lo referente a los objetivos del Trabajo de Final de Master (TFM), el desarrollo del presente documento ha permitido al autor profundizar en lo referente a la aplicación del *blockchain* en la gestión soberana de la identidad. A pesar de que mucha de información está publicada ya en la Red, cierto es que no existe bibliografía en español y que la existente se encuentra muy dispersa, centrándose más en las herramientas que en el concepto de gestión soberana.

El Trabajo queda como punto de partida para una futura ampliación de conceptos técnicos, que quien sabe, podrían contribuir en el propio proyecto *Hyperledger Indy*. Referente a estos conceptos más técnicos no existe a día de hoy información en la Red, dado el grado de madurez del proyecto. Ejemplo de estas carencias seria todo lo relativo a la criptografía utilizada, tanto para verificar las evidencias y los *claims* genéricos como para los *Zero Knowledge Proofs*. Continuación de este Trabajo podría ser un análisis de las funciones criptográficas necesarias para implementar las funciones propias del SDK, reforzando y divulgando pedagogía sobre la Criptografía de Curva Elíptica (ECC).

7. GLOSARIO

Se definen a continuación de los términos y los acrónimos más relevantes.

Término	Definición
CA	[*Certification Authority*] Entidad que, en el contexto de la infraestructura de clave pública (PKI) certifica, mediante un certificado digital, la propiedad de una clave pública.
Claim	[*Claim record*] En el contexto de Sovrin, un tipo de *Identity record* que asevera uno o más atributos de un *Identity owner*.
Credential	En el contexto de Sovrin, *Idenentity record* que asevera (en un único registro) más de un atributo de un *Identity Owner*.
Credential Definition	En el contexto de Sovrin, *schema* instanciado por parte de un *Issuer* concreto, y que será utilizado para generar los *claims*.
DID	[*Descentralized IDentifier*] En el contexto de Sovrin, identificador global único (UUID) vinculado a un par de claves público-privada que se usa para identificar a los actores de la *Sovrin Network*.
DLT	[*Distributed Ledger Technology*] Término utilizado para referirse de manera genérica a todas las tecnologías que implementan una base de datos descentralizada.
DPKI	[*Descentralized* o *Distributed Public Key Infrastructure*] Infraestructura de clave pública en la que la propiedad de las claves públicas no se atestigua mediante una CA sino mediante una base de datos distribuida basada en *blockchain*.
Docker	Herramienta, fruto de un proyecto de código abierto, que permite automatizar el despliegue de aplicaciones mediante contenedores lógicos que resuelven todas las dependencias que estas puedan tener (S.O., drivers, MW, etc.).
Evernym	Empresa pionera en el desarrollo de una *blockchain* orientada a solucionar los problemas asociados a la gestión soberana de la identidad digital.
Hyperledger Indy	Proyecto de código abierto iniciado por Evernym y orientado a implementar un ecosistema técnico para la gestión soberana de identidades basado en tecnología *blockchain*.
IDE	[*Integrated Development Environment*] Conjunto de herramientas integradas orientadas al desarrollo de aplicaciones de software.

Término	Definición
Idenity register	En el contexto de Sovrin, registro escrito en la Sovrin *ledger* a través de una *Identity transaction* y que sirve para conformar la identidad digital de los *Identity Owners*.
Identifier record	En el contexto de Sovrin, tipo específico de registro escrito en el Sovrin *ledger* que sirve para registrar los DID del sistema.
Identity Owner	En el contexto de Sovrin, entidad descrita por la *Sovrin identity*, pudiendo ser un individuo, una organización o una cosa.
IDMS	[*IDentification Management System*] Sistema de información que permite almacenar y administrar los atributos de un conjunto de usuarios gestionar su autenticación y posteriormente regular las reglas de acceso de estos a distintos recursos.
IdP	[*Identity Provider*] Componente de un sistema que genera, mantiene y gestiona la información vinculada a la identidad para proveer un servicio de autenticación.
Indy Plenum	Protocolo que caracteriza a la *blockchain* vinculada al proyecto *Hyperledger Indy* y que se fundamenta en el uso de unos mecanismos basado en RBFT para garantizar la disponibilidad de la infraestructura.
JSON	[*JavaScript Object Notation*] Formato de texto utilizado para el intercambio de datos basado en pares (tuplas) de claves y valores.
KYC	[*Know Your Customer*] Procesos aplicados principalmente en el entorno bancario que regulan la obtención de datos de los clientes antes de poder ofrecerles servicios financieros.
Ledger	Término utilizado para referirse a la base de datos de una *blockchain*, en la que se registra información que puede considerarse inmutable.
Onboarding	En el contexto de Sovrin, proceso que permite establecer una conexión entre dos actores. El objetivo del proceso es que ambas partes de autentiquen de manera segura.
P2P	[*Peer to Peer*] Comunicación que se establece entre dos equipos asumiendo ambos el mismo rol (no existe una jerarquía entre ellos ni un rol diferenciado como podría ser el de cliente y servidor).
PKI	[*Public Key Infrastructure*] Entorno tecnológico que permite la ejecución de operaciones criptográficas (encriptación, firmado digital, autenticación, etc.) entre los distintos participantes fundamentándose en los principios de la criptografía asimétrica.

Término	Definición
Proof Request	En el contexto de Sovrin, condiciones que establece un *Verifier* para que un *Identity Owner* demuestre de manera fehaciente determinados atributos.
Provisioning	En el contexto de Sovrin, proceso que se ejecuta cuando se da de alta una nueva identidad en la *Sovrin Network*.
Pseudonym	En el contexto de Sovrin, DID que sirve para establecer una relación con otro actor del sistema y que no identifica de manera inequívoca a su propietario.
RBFT	[*Redundant Byzantine Fault Tolerance*] Protocolo que, en una red de nodos, regula el establecimiento de un nodo principal y del resto como seguidores. Fundamentalmente se basa en la monitorización del rendimiento del nodo principal, iniciándose un reemplazo de este cuando su rendimiento cae por debajo de un determinado umbral.
RP	[*Relaying Party*] Componente de un sistema que requiere de un proceso previo de autenticación para permitir el acceso a los usuarios.
Schema	En el contexto de Sovrin, esquema de datos que define una evidencia o *claim*.
SDK	[*Software Development Kit*] Conjunto de herramientas de desarrollo software que permiten al programador crear una aplicación para un entorno concreto.
Verinym	En el contexto de Sovrin, DID que permite identificar de manera inequívoca a un actor del sistema.
Wallet	En el contexto de Sovrin, contendor lógico dónde los Clientes almacenan básicamente lo relativo a sus claves y a sus evidencias.
ZKP	[*Zero Knowledge Proof*] Evidencia que, mediante unos mecanismos criptográficos específicos, permite revelar de manera selectiva únicamente determinada información de nuestra identidad.

Tabla 2. Glosario de términos y acrónimos

[página intencionadamente en blanco]

8. BIBLIOGRAFÍA

Artemkaaas, & varios, a. (7 de Noviembre de 2018). *GitHub Hyperledger Indy-SDK*. Obtenido de Indy-SDK GitHub website: https://github.com/hyperledger/indy-sdk/blob/master/doc/getting-started/getting-started.md

Aublin, P.-L., Mokhtar, S. B., & Quéma, V. (26 de Julio de 2016). *Grenoble University, CNRS-LIRIS, Grenoble INP.* Obtenido de http://lig-membres.imag.fr/aublin/rbft/report.pdf

Evernym. (29 de Septiembre de 2016). *Sovrin Glossary*. Obtenido de https://www.evernym.com: https://www.evernym.com/wp-content/uploads/2017/07/Sovrin-Glossary.pdf

Evernym. (29 de Septiembre de 2016). *Sovrin Technical Foundations*. Obtenido de https://www.evernym.com: https://www.evernym.com/wp-content/uploads/2017/07/The-Technical-Foundations-of-Sovrin.pdf

Sovrin Foundation. (3 de Octubre de 2016). *How Sovrin Works*. Obtenido de https://sovrin.org: https://sovrin.org/wp-content/uploads/2018/03/How-Sovrin-Works.pdf

Sovrin Foundation. (28 de Junio de 2017). *Sovrin Provisional Trust Framework*. Obtenido de https://sovrin.org: https://sovrin.org/wp-content/uploads/2018/03/Sovrin-Provisional-Trust-Framework-2017-06-28.pdf

World Economic Forum. (Agosto de 2016). *World Economic Forum.* Obtenido de Blueprint for Digital Identity: http://www3.weforum.org/docs/WEF_A_Blueprint_for_Digital_Identity.pdf