

**DOCUMENT MADE AVAILABLE UNDER THE
PATENT COOPERATION TREATY (PCT)**

International application number:	PCT/EP2020/059280
International filing date:	01 April 2020 (01.04.2020)
Document type:	Certified copy of priority document
Document details:	Country/Office: US
	Number: 62/827,626
	Filing date: 01 April 2019 (01.04.2019)
Date of receipt at the International Bureau:	22 April 2020 (22.04.2020)

Remark: Priority document submitted or transmitted to the International Bureau in compliance with Rule 17.1(a),(b) or (b-bis)

CERTIFICATE OF AVAILABILITY OF A CERTIFIED PATENT DOCUMENT IN A DIGITAL LIBRARY

The International Bureau certifies that a copy of the patent application indicated below has been available to the WIPO Digital Access Service since the date of availability indicated, and that the patent application has been available to the indicated Office(s) as of the date specified following the relevant Office code:

Document details: Country/Office: US

Filing date: 01 Apr 2019 (01.04.2019)

Application number: 62827626

Date of availability of document: 10 Apr 2019 (10.04.2019)

The following Offices can retrieve this document by using the access code:

AR, AU, BR, CA, CL, CN, DK, EA, EE, EP, ES, FI, GB, GE, IB, IL, IN,
JP, KR, MA, NL, NO, NZ, SE, US

Date of issue of this certificate: 22 Apr 2020 (22.04.2020)



THE UNITED STATES OF AMERICA

TO ALL TO WHOM THESE PRESENTS SHALL COME:

UNITED STATES DEPARTMENT OF COMMERCE

United States Patent and Trademark Office

April 22, 2020

THIS IS TO CERTIFY THAT ANNEXED HERETO IS A TRUE COPY FROM THE RECORDS OF THE UNITED STATES PATENT AND TRADEMARK OFFICE OF THOSE PAPERS OF THE BELOW IDENTIFIED PATENT APPLICATION THAT MET THE REQUIREMENTS TO BE GRANTED A FILING DATE UNDER 35 USC 111.

APPLICATION NUMBER: 62/827,626

FILING DATE: *April 01, 2019*

THE COUNTRY CODE AND NUMBER OF YOUR PRIORITY APPLICATION, TO BE USED FOR FILING ABROAD UNDER THE PARIS CONVENTION, IS *US62/827,626*



Certified by

Andres Ibarra

Under Secretary of Commerce
for Intellectual Property
and Director of the United States
Patent and Trademark Office

DocCode – SCORE

SCORE Placeholder Sheet for IFW Content

Application Number: 62827626

Document Date: 04/01/2019

The presence of this form in the IFW record indicates that the following document type was received in electronic format on the date identified above. This content is stored in the SCORE database.

Since this was an electronic submission, there is no physical artifact folder, no artifact folder is recorded in PALM, and no paper documents or physical media exist. The TIFF images in the IFW record were created from the original documents that are stored in SCORE.

- Drawing

At the time of document entry (noted above):

- USPTO employees may access SCORE content via eDAN using the Supplemental Content tab, or via the SCORE web page.
- External customers may access SCORE content via PAIR using the Supplemental Content tab.

Form Revision Date: August 26, 2013

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re **PROVISIONAL** Patent Application of: Fabrice Jogand-Coulomb et al.

Title: SYSTEM AND METHOD FOR PROVIDING AGGREGATED CREDENTIALS WITH ASSURANCE
LEVELS

Docket No.: 5483.145PRV

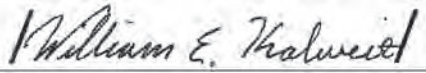
Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

We are transmitting herewith the following attached items (as indicated with an "X"):

- ☒ Provisional Application Cover Sheet (1 pg)
- ☒ Authorization to Permit Access (PTO/SB/39) (1 pg)
- ☒ PROVISIONAL Patent Application comprising:
 - ☒ Specification with Color Drawings throughout (98 pgs)
- ☒ Authorization to charge Deposit Account 19-0743 in the amount of **\$280.00** to cover the provisional filing fee and any applicable application size fee.

Please charge any additional required fees or credit overpayment to Deposit Account No. 19-0743.

SCHWEGMAN LUNDBERG & WOESSNER, P.A.
Customer Number: 159468

By: 
William E. Kalweit
Reg. No. 70,481

Date of Deposit: April 1, 2019

This paper or fee is being filed on the date indicated above using the USPTO's electronic filing system EFS-Web, and is addressed to The Commissioner for Patents, P.O. Box 1450, Alexandria, VA 22313-1450.

PROVISIONAL APPLICATION FOR PATENT COVER SHEET

This is a request for filing a PROVISIONAL APPLICATION FOR PATENT under 37 CFR 1.53(c).

INVENTOR(S)			
Given Name (first and middle (if any))		Family Name or Surname	Residence (City and either State or Foreign Country)
Fabrice Jean-Baptiste		Jogand-Coulomb Milan	Aix en Provence, France Boosterstown, Ireland
TITLE OF THE INVENTION (500 characters max):			
SYSTEM AND METHOD FOR PROVIDING AGGREGATED CREDENTIALS WITH ASSURANCE LEVELS			
CORRESPONDENCE ADDRESS			
Direct all correspondence to: The address corresponding to Customer Number:		159468	
ENCLOSED APPLICATION PARTS (check all that apply)			
	Application Data Sheet. See 37 CFR 1.76		CD(s), Number of CDs: _____
	Drawing(s) Number of Sheets	X	Other (specify): <u>Authorization to Permit Access (PTO/SB/39) (1 pg.)</u>
X	Specification with Color Drawings throughout Number of Pages (98)		
Fees Due: Filing Fee of \$280 (\$140 for small entity) (\$70 for micro entity). If the specification and drawings exceed 100 sheets of paper, an application size fee is also due, which is \$400 (\$200 for small entity) (\$100 for micro entity) for each additional 50 sheets or fraction thereof. See 35 U.S.C. 41(a)(1)(G) and 37 CFR 1.16(s).			
METHOD OF PAYMENT OF THE FILING FEE AND APPLICATION SIZE FEE FOR THIS PROVISIONAL APPLICATION FOR PATENT			
	Applicant asserts small entity status. See 37 CFR 1.27.		
	Applicant certifies micro entity status. See 37 CFR 1.29. <i>Applicant must attach form PTO/SB/15A or B or equivalent.</i>		
	A check or money order made payable to the <i>Director of the United States Patent and Trademark Office</i> is enclosed to cover the filing fee and application size fee (if applicable).		
	Payment by credit card. Form PTO-2038 is attached		
X	The Director is hereby authorized to charge the filing fee and application size fee (if applicable), as well as any additional required fees, or credit any overpayment to Deposit Account Number: 19-0743.		
TOTAL FEE AMOUNT \$ \$280.00			
The invention was made by an agency of the United States Government or under a contract with an agency of the United States Government.			
X	No.		
	Yes, the invention was made by an agency of the U.S. Government. The U.S. Government agency name is:		
	Yes, the invention was made under a contract with an agency of the U.S. Government. The name of the U.S. Government agency and Government contract number are:		

SIGNATURE:



Date: April 1, 2019

TYPED or PRINTED NAME: William E. Kalweit

REGISTRATION NO. 70,481

TELEPHONE: (612) 359-3266

DOCKET NUMBER: 5483.145PRV

UNITED STATES PROVISIONAL PATENT APPLICATION

**SYSTEM AND METHOD FOR PROVIDING AGGREGATED
CREDENTIALS WITH ASSURANCE LEVELS**

INVENTORS

Fabrice Jogand-Coulomb
Jean-Baptiste Milan

Schwegman Lundberg & Woessner, P.A.
1600 TCF Tower
121 South Eighth Street
Minneapolis, MN 55402

Attorney Docket Number: 5483.145PRV
Client Ref. No. PA-200-0009.US

SYSTEM AND METHOD FOR PROVIDING AGGREGATED CREDENTIALS WITH ASSURANCE LEVELS

TECHNICAL FIELD

[0001] Embodiments described herein generally relate to electronic or digital credential systems, and in particular, to aggregating, verifying, and issuing personal credentials.

BACKGROUND

[0002] Identity documents, such as identity cards, may have multiple pieces of information stored on them. Some countries have begun a move to national identification cards (e.g., ePassports) that include a microchip or integrated circuit that electronic stores information. A national identification card may include basic information, such as a person's name, home address, date of birth, age, or gender. Additionally, a national identity card may include security mechanisms such as an electronic security certificate, anti-counterfeit printing techniques, or seals.

BRIEF DESCRIPTION OF THE DRAWINGS

[0003] In the drawings, which are not necessarily drawn to scale, like numerals may describe similar components in different views. Like numerals having different letter suffixes may represent different instances of similar components. Some embodiments are illustrated by way of example, and not limitation, in the figures of the accompanying drawings in which:

[0004] FIG. 1 is a block diagram illustrating a credential system, according an embodiment;

[0005] FIG. 2 illustrates a data structure and naming conventions of a data element, according to an embodiment;

[0006] FIG. 3 is a diagram illustrating package data structures, according to an embodiment;

[0007] FIG. 4 is an example aggregated data structure used to store data from various issuers, according to an embodiment;

[0008] FIG. 5 is a message sequence diagram illustrating electronic credential generation and issuance process, according to an embodiment;

[0009] FIGS. 6 and 7 are example user interfaces, according to embodiments;
[0010] FIG. 8 is a flowchart illustrating a method of installing and registering an aggregated credential user application, according to an embodiment;
[0011] FIG. 9 is a block diagram illustrating an example machine upon which any one or more of the techniques (e.g., methodologies) discussed herein may perform, according to an embodiment.

DETAILED DESCRIPTION

[0012] In the following description, for purposes of explanation, numerous specific details are set forth in order to provide a thorough understanding of some example embodiments. It will be evident, however, to one skilled in the art that the present disclosure may be practiced without these specific details.

[0013] While various electronic identification cards have been proposed, they fail to provide assurances to verify the authenticity and validity of the information stored on the card. Additionally, there is the risk of losing a card and as a result, losing an identification credential. These shortcomings, along with others, are addressed in the system described herein.

[0014] The present disclosure describes an improved credential system that provides a portable, electronically verifiable, and aggregated credential. Identity credentials (e.g., documents, cards, mobile profiles, etc.) include information such as a first and last name, a date of birth, an address, etc. In the digital world, each piece of information is a data element and these data elements are combined into a package that is issued to a target user. The aggregated credential brings together information from multiple issuers rather than duplicating the information. For example, if an insurance credential is issued to a user based on a government identification credential, the aggregated insurance credential references into the government identification credential to supply the user's name rather than copying the name into the insurance credential. This reference to the government credential avoids data duplication and the attendant synchronization problems to keep information up-to-date across many different issuers of information.

[0015] Having a digital credential provides several advantages. Information and documentation are easy to access. The information, or reference to the information, is centralized in the aggregated credential. Not only is this

convenient, but reference to issuers also helps to ensure that the information is current. Digital credentials may also be more immune to counterfeiting and corruption than conventional credentials. These features and more are discussed below.

[0016] The present disclosure is organized as follows:

[0017] 1) System Overview

[0018] 2) Package Data Structure

[0019] 3) Package Issuance

[0020] 4) Package Updates and Deletion

[0021] 5) User Application

[0022] 6) Verifier Operations

[0023] Some terms are provided here for common reference. It is understood that these terms are non-limiting and that other terms, phrases, or descriptions of operations, components, or devices in the systems and methods described may be used.

[0024] Package – a Package is a, possibly empty, collection of name:value pairs of data. The package can also include meta-data, such as commands, preferences, etc. The name of the data pair may be referred to as a Tag, Field, or Description. The value may be an actual value or a reference to an actual value. The reference may be directed to another package, another name:value pair within the same package, or elsewhere. A Package may include metadata about the Package, such as a Package Hash, Package Signature, Package Expiration, or the like.

[0025] Credential – a Credential is an encapsulation of one or more packages. An aggregated credential includes two or more packages.

[0026] Issuer – an Issuer is a person, entity, service, or other platform that provides a Package. The Issuer may attest to the authenticity, validity, and correctness of data contained in the Package. The Issuer may sign the Package or sign individual data in the Package. The Issuer may be an original issuer, which is an Issuer that created the data, or a re-issuer, which is one that reuses the Original Issuer's data. The original issuer issues a stand-alone (e.g., not aggregated) package because they do not reference or import other data. An example of a stand-alone package could be a birth certificate. The re-issuer provides an aggregated package, whereby at least one name:value pair is a

reference to another package, which can be either a stand-alone package or another aggregated package.

[0027] Data that is provided by an Issuer may have an associated Assurance Level. An Assurance Level is an indication of the trustworthiness of the data. The Assurance Level may be a numerical range, a letter grade, or other indication. Data issued by an Original Issuer may have the highest Assurance Level. As data is incorporated by reference by Re-Issuers in derivative Packages, the Assurance Level may decrease stepwise. By analyzing the Assurance Level of data contained in a Package, one may be able to see how many steps away the data in the current Package is from the Original Issuer.

[0028] Verifier – a Verifier is a person, entity, service, or other platform that verifies data in a Package or Credential. A Verifier may obtain data from a user Credential, for example a point-of-sale, and then contact one or more Issuers to validate the data. Other forms of validation may be used, such as by analyzing data signatures, analyzing a blockchain, or the like. For example, data can be signed by an original issuer. The signature can be presented by the user (e.g., holder of the credential) at a point-of-sales point to collect and verify the user's information. The verifier can receive the data from the Package and check the various issuer signatures for any package referenced to establish a trust level for the various issuers.

System Overview

[0029] FIG. 1 is a block diagram illustrating a credential system 100, according an embodiment. The credential system 100 includes a first original issuer 102A and a second original issuer 102B (collectively referred to as 102) that issue core packages 104A, 104B (collectively referred to as 104). The core packages 104 include tags and values that are generated by the original issuers 102. The credential system 100 also includes re-issuers 106A, 106B (collectively referred to as 106) that compile data from one or more issuers (or re-issuers) in the credential system 100 and re-issues an aggregated credential with reference to one or more packages.

[0030] A re-issuer 106 may request specific data from an issuer 102, to populate a name:value pair. For example, the request may include a minimum assurance level. Specifying a minimum assurance level rather than specifying a

particular issuer 102, allows data to be referenced from multiple issuers 102, who may have data with an adequate level of assurance. Instead of a minimum, as an alternative, the request may specify an assurance level. In some examples, the re-issuer 106 may check the value and in some other cases the values do not matter and only the minimum assurance level or issuer is sufficient. For example, it may not matter if the name of the person is John or Patrick as long as the name is from a trusted identity. Such process can then work for a verification: the verifier requests the tag(s) and then receives the response. The tags could be for the value (received with a reference) or for the reference only.

[0031] The re-issuer 106 may also add its own data to the aggregate credential by issuing its own package and consolidating the package with data from packages of other issuers. The re-issuer 106 may sign its own package, portions of its own package, or the aggregated credential.

[0032] When preparing a package that contains data from one or more other packages, the re-issuer 106 may check the referenced data. The re-issuer 106 may verify the referenced data by interfacing with the original issuer 102, checking parameters (e.g., data validity/expiration), reviewing a blockchain, or the like. In some examples, the original issuer is not involved in the verification, but rather the re-issuer checks the signature to determine whether it is from a trusted authority.

[0033] In the example illustrated in FIG. 1, the first original issuer 102A is a government agency that issues birth certificates (e.g., a Public Health Office of a county). The first original issuer 102A is the place where the user's name, date of birth (DOB), and place of birth are first recorded. The second original issuer 102B is a vehicle manufacturer that produces vehicles. Each vehicle produced is uniquely identified with a vehicle identification number (VIN). The second original issuer 102B creates a package with data describing a vehicle.

[0034] The re-issuer 106A may be a government agency that produces packages for vehicle titles (e.g., a Department of Motor Vehicles (DMV)). The vehicle title is identified with a title identifier that uniquely identifies the title. Consequently, the aggregate credential produced by the re-issuer 106A may include: the vehicle title identifier, as issued by the re-issuer 106A; a reference to a package from first original issuer 102A describing an owner's personal

information; and possibly a reference to a second package from the second original issuer 102B describing the vehicle.

[0035] A reference may be stored in the aggregated credential that points to data from another package. The reference may be a Uniform Resource Identifier (URI) or Uniform Resource Locator (URL). When data is reproduced and stored local to the aggregate credential, the assurance level may be decreased to reflect the possibility that the value is no longer correct or the issuer may not be a legitimate source for such information. When data is referenced instead, then the assurance level may remain at the same level as that in the referenced package.

[0036] The re-issuer 106B may be an insurance company that provides motor vehicle insurance. The re-issuer 106B may produce a policy number that identifies an insurance policy for the vehicle. The package that includes the policy number may also include, using actual values or references to the actual value, information about the insured party (e.g., data from package created by issuer 102A), information about the vehicle being insured (e.g., data from package created by issuer 102B), and information about the vehicle title (e.g., data from aggregate credential produced by re-issuers 106A).

[0037] A credential database 108 may be used to store packages and other data from one or more issuers 102 or re-issuers 106. The credential database 108 may use a relational database management system (RDBMS) to organize the package information into tables, may be organized as a NoSQL data store, a filesystem, or other configuration to hold and find packages. The credential database 108 may be queried by various entities or users, such as a re-issuers 106, a verifier 110, or a user at a user device 112. A re-issuer 106 may query the credential database 108 to obtain original package information (e.g., information contained within packages or information about packages) to populate an aggregated credential. The credential database 108 may be on the user device 112, a verifier 110, implemented on one or more servers, which may be owned or operated by a credential issuing entity, or other convenient location. In some embodiments, credential issuing entities access the credential database 108 as part of a service.

[0038] The various components of the credential system 100 may communicate over one or more networks 114, which may include any known type of network that facilitates machine-to-machine communications. The

network 114 may use the same communication protocols or different protocols without departing from the scope of the present disclosure. The network 114 may include wired or wireless communication technologies. The Internet is an example of a communication network that constitutes an Internet Protocol (IP) network consisting of many computers, computing networks, and other communication devices located all over the world, which are connected through many telephone systems and other means. Other examples of a communication networks include, without limitation, a Plain Old Telephone System (POTS), an Integrated Services Digital Network (ISDN), the Public Switched Telephone Network (PSTN), a Local Area Network (LAN), a Wide Area Network (WAN), a Session Initiation Protocol (SIP) network, a Voice over Internet Protocol (VoIP) network, a cellular network, and any other type of packet-switched or circuit-switched network known in the art. In addition, it can be appreciated that the communication network 114 need not be limited to any one network type, and instead may be comprised of several different networks or network types. Moreover, the network 114 may include a number of different communication media such as coaxial cable, copper cable/wire, fiber-optic cable, antennas for transmitting/receiving wireless messages, and combinations thereof.

[0039] The user device 112 and verifier device 110 may be of any type of compute device. The user device 112 is typically portable in nature, and may take the form of a cellular phone, mobile device, smart phone, personal digital assistant, laptop, tablet, wearable device, portable credential card, key fob, or the like. It should be appreciated that the verifier device 110 does not necessarily have to correspond to a mobile device, but rather may correspond to a personal computer, desktop computer, kiosk, payment terminal, or the like.

[0040] During a transaction, a user with the user device 112 may be asked to present a credentials to a verifier device 110. The verifier device 110 may be a point-of-sale terminal with a beacon. The beacon broadcast may be used to periodically attempt to connect with user devices 112 in communication range or within some threshold proximity. In some examples, the beacon broadcast can be used to wake an application subscribed to changes in the information. The verifier device 110 may connect with the user device 112 over wired or wireless communications (e.g., BLUETOOTH® or WI-FI), and obtain some or all of a credential from the user device 112. For instance, as the user approaches the

checkout lane, the beacon (e.g., two-way Bluetooth Low Energy (BLE) device of the like) may advertise a connection, which the user device 112 uses to connect with the beacon and construct a secure connection. The beacon may request certain information, such as a user's name and credit card number, and the user device 112 may respond with the information. The user may selectively share data elements from one or more packages to the verifier device 110 to maintain as much personal privacy as possible. In some examples, the verifier device 110 may validate the credential by accessing an issuer 102, re-issuer 106, or credential database 108. In some examples, the verifier device 110 does not access the issuer 102, the re-issuer 106, or the credential database 108 (e.g., unless it is local to the verifier device 110), but rather performs the verification by checking the issuer signature and confirming that the issuer is trusted.

Package Data Structure

[0041] FIG. 2 illustrates a data structure and naming conventions of a data package 200, according to an embodiment. The data package 200 includes several data elements 202, each identified with a "tag." The tag 204 may also be referred to as a field name or a name. The data elements of FIG. 2 include a tag of "first_name" and is used to store the first name of the user. Additional tags of "last_name" and "gender" are included in the example shown. Tags may be formatted using various conventions, such as with underlines, colons, dashes, or other characters to separate namespaces and tags.

[0042] A namespace is a logical grouping of tags. In some examples, the namespace defines a context for the tags to ensure understanding by all parties. The namespace may be at the package level as is illustrated in FIG. 2. As an alternative, the namespace may be at the data element level. For instance, a namespace of "personal_info" may be used for the tags "first_name" and "last_name" and be labeled as "personal_info:first_name" and "personal_info:last_name." It is understood that the naming convention here for namespaces and tags is merely for illustration and that other naming conventions may be used.

[0043] Each tag has an associated value 206. The value may also be referred to an attribute or value field. The value for the "first_name" tag is "Vincent." Additional values for other tags are illustrated as examples. Values may be of

any data type including, but not limited to numerals, character strings, alphanumeric strings, floats, longs, or the like. Values may be constrained using maximum length, minimum length, maximum value, minimum value, or the like.

[0044] Additional tags may be used to capture attributes be associated with a tag. For instance, the “first_name_label” tag is an attribute of the “first_name” tag, and has a value “FIRST NAME.” The “first_name_label” attribute tag may be used by an application to generate a user interface with the provided value. The details of this data structure are largely left to implementation preferences. In some examples, a tag reference (tag_ref) can be used to request a reference to a tag without receiving the data that corresponds to the tag.

[0045] Additional tags may be used to capture the datatype of a tag. For instance, the “date_of_birth” tag may have an associated datatype tag of “date_of_birth_datatype” with a value of “DATE.” This datatype tag may be used by a user interface to display the value of the “date_of_birth” tag with appropriate formatting. Other types of datatypes may indicate the MIME type, mathematical precision, image size, or other constraint on the formatting of the associated tag.

[0046] Tag and values may refer to literal values or referenced values. A literal value is a string or other value that is stored in the value field as an actual value. For instance, a literal value of the tag “first_name” may be “Sarah.” In contrast, a referenced value is a string or other value that refers to another location where the value of the value field is stored. For instance, a referenced value of the tag “last_name” may be “www.issuer_a.id/Ab35H/38FE”, which refers to a package “Ab35H” with hash “38FE” and includes the value of the “last_name” field. Discussion of package names, hashes, and how they are accessed is discussed further below.

[0047] As such, a simple package includes zero or more tag/value pairs (e.g., data elements), where each tag is associated with a namespace. A simple package may be empty, so it may have no tag/value pairs. After being instantiated the simple package may be populated with one or more tag/value pairs, as is discussed further below. The tags in a simple package may refer to another tag in the same package or a tag from a different package. Tags stored in a package may refer to data elements from the same namespace or from different

namespaces. Further, there may be multiple of the same type of tags in a given package.

[0048] A simple package also optionally includes several pieces of metadata that describe the package. This metadata is stored in the package as tag/value pairs. Examples of metadata may include package issuer, issuer electronic signature, package hash value, access control data, location-based services data, expiration date, version number, or the like.

[0049] An aggregated package includes one or more references to data from at least one other package. An aggregated package may also include metadata such as the package issuer, issuer electronic signature, package hash value, access control data, location-based services data, expiration date, version number, or the like. In some cases, an aggregated package may not be associated with an issuer. For instance, an aggregated package may only include tags and values from other issuers. In other cases, an aggregated package is created from a simple package, where a simple package is used as the base package and additional package data is added to it. An aggregated package may reference one or more data elements another aggregated package. Aggregated packages may be stored at the credential database 108 or at a re-issuer 106.

[0050] An aggregated package is used by an application executing on a mobile device (e.g., user device 112). The application is able to access the aggregate package and retrieve information from the aggregate package to share with a verifier (e.g., verifier device 110) or to present to the user. In some examples, an application presents information to another as selected by an end user. In some examples, the application receives a request and responds with the information. Additional functionality of applications is discussed further below.

[0051] FIG. 3 is a diagram illustrating package data structures, according to an embodiment. In FIG. 3, standalone packages 300 A and 300 N may have some or all of their data to be reference in one or more aggregated packages, such as aggregated package 1 302 and aggregated package 2 304. Each package 300A-N and aggregated package 302, 304, may be network-accessible using a URI.

[0052] To reference data elements from a package, the URI of the package may be expanded with a reference of the data element, its position in the package, and a hash of the data element or the package the data element belongs

to. Keeping track of the hash allows the requester to detect if a package or data element has changed.

[0053] As discussed above, the tag may be associated with a value that is either an actual value or a reference to the data element associated with the value. Tags may also be associated with other attributes, such as labels and datatypes, which may be used in a user interface to format or configure presentation.

[0054] When a value is referenced with a URI, the URI may have the following format: [download URL]/[package ID]/[package hash]/[tag ID]. The download URL may be a network URI that uses the HTTP protocol. The package ID may be a unique identifier for the package. The tag ID may be an index into the package, a unique identifier for the tag, the tag name, or some other indicia of the tag in the package. Using a URI provides a simple way for all necessary entities to access the package and the referenced data within.

[0055] An example of a standalone package 300 is a government citizen identity. Another example is a vehicle registration before the vehicle is sold to an owner.

[0056] An aggregated package 302, 304 may be anything that references a data element (e.g., name, access rights, a receipt from a sale, etc.) in another package, such as a driving license, health coverage, flight ticket, etc. An aggregated package may also be used for things like payment credentials, which allows a verifier to check a buyer's identity or age information in the same transaction as payment, access control data so that biometrics of the person identity may be verified during access control, etc.

[0057] In some case, the aggregated credential is not for specific data elements but for a document. In this case, a new credential may be bound to an issued document. This may be useful for example to bind a credential used for access control to an identity document. In such a process the package matching the target kind of document is first identified using typical aggregated credential issuance described further down. Then credentials are issued and aggregated to the identified package.

[0058] While such binding is not strongly enforced by the application or based on something the user owns, binding a credential to a document allows the system to bind a credentials that could be used by other kinds of verifiers to a

document. It also is a simple mechanism to ensure such aggregated credentials may be rendered obsolete when the package it is aggregated to changes or expires. Further the referencing method may be similar to the one for a data element, but the URI is shortened to only reference the package. For example, the URI may be in the form of: [download URL]/[package ID]/[package hash]. The enforcement is then handled by the application itself and could rely on policies delivered along with the package.

[0059] FIG. 4 is an example aggregated data structure 400 used to store data from various issuers, according to an embodiment. The aggregated data structure 400 may comprise any data structure suitable for storing data and issuing data in accordance with at least some embodiments of the present disclosure. As illustrated, the data element references in the data structure 400 have the following correspondence to the packages 402: “DL” is the DL package; “SID” is the student-ID package; and “HC” is the health care package.

[0060] The aggregated data structure 400 may compile data packages 402A-N from various issuers. In some examples, the packages can be organized by namespace. Examples of issuers include, but are not limited to, governmental entities, healthcare providers, educational entities, travel services, financial institutions, utility providers (e.g., cable, electric, cell phone providers, etc.). In this example, data package 402A comprises driver’s license data received from the DMV, healthcare package 402B comprises coverage information received from a healthcare insurer, student ID package 402C is received from an educational entity, and boarding pass package 402N is received from an airline.

[0061] The driver’s license package 402A includes the data tags: name, driver’s license number, date of birth, address, height, weight, photo, issued data, and expiration date. The healthcare package 402B includes the data tags: name, date of birth, address, plan number, group number, member number, issued date, and expiration date. The student ID package 402C includes the data tags: name, date of birth, address, student ID number, issued date, and expiration date. The boarding pass package 402N includes the data tags: name, flight number, departure date, departure time, departure city, arrival airport/city, arrival time, and seat number.

[0062] As shown, all packages 402A-N include the data tag name. Data packages 402A-N include the date of birth (DOB) tag. Therefore, if a new issuer

wants to reference the name or DOB data tag, there are multiple issuers that may be referenced. The referenced data may be chosen based on factors such as required assurance level, associated cost, verification status, expiration date, etc. Although, not shown, each data package 402A-N may include a unique identifier. Data packages 402A-N are for illustration only, and it is understood that in some implementations, packages 402A-N may include additional data tags, fewer data tags, different data tags, or differently arranged data tags than those that are shown in FIG. 4.

[0063] In some embodiments, a data package may include or reference a given data tag multiple times. For example, the identical tag may come from different people associated with the same package, such as all individuals covered under the same car insurance policy. Therefore, the data tag and corresponding value for the policy number may be identical across all individuals covered under the same policy. In another example, the same data value may be used multiple times in the same package. For example, the insurance package 402B may indicate a primary member of the insurance package 402B and also covered member(s), where the individual listed as the primary member may also be listed under covered member(s).

[0064] In some embodiments, aggregating credentials may result in a data tag being issued and used only once for a given individual. All issued tags for a given individual may be stored in a single table and the table may contain a unique instance of each data tag. Other tables may be used to manage data packages to indicate where the data tag is referenced. In other embodiments, a table may store a data tag along with a reference to the issuer of the data, allowing multiple versions and update history to be tracked. In yet other embodiments, where level of assurance is considered, the same data tag may be used multiple times by various issuers.

Package Issuance

[0065] In general, a package is issued at the request of an application that is executing on the user's user device. The user may initiate the request of an electronic credential through the use of an application. The application may be associated with a service, entity, or other platform. For instance, an application may be associated with an insurance company and the user may access the

application to request a proof-of-insurance credential. As another example, an application may be associated with a government agency and the user may initiate a request to obtain a proof-of-identity credential. Other applications and associations are understood to be within the scope of this disclosure.

[0066] Depending on the type of issuer (e.g., original issuer or re-issuer), the package obtained by the user at the user device may be a standalone package or an aggregated package. Before an aggregated package may be issued, at least one standalone package is issued. Then the aggregated package may refer to one or more data elements from the standalone package(s) and include references to the values of the standalone package(s) in the aggregated package.

[0067] Issuing a standalone package typically only involves a single issuer. The issuer may restrict the kind of application that is able to receive a package issued by the issuer. Many methods are available for a remote system to identify an application. These methods are not described here. However, an authorized issuer may be provided with the necessary information to send notification to that particular application instance, whether directly to the application or through some application backend, indicating whether the application is suitable or authorized to receive the package.

[0068] The package may be issued to the application in several ways. The actual mechanism depends in part on the desired user experience. For example, if the request is performed from within the target application, then it is easy to collect the details of the application. However, when the trigger is received by the target user, independent from the target application (e.g., some information that triggers the acquisition is provided to the target application such as an email with a magiclink or deeplink to the application with the provisioning information), then the optional identification process starts only after the acquisition trigger is received. There are also cases where the target application is known but the instance is not.

[0069] Though various mechanisms, at some point the selected application from the user initiates the acquisition. Here again various cases may apply and more particularly the package may be delivered through the application backend (e.g., stored at a network location accessible by the application) or directly from the issuer to the application (e.g., stored locally in memory accessible by the application). While acquiring through the application backend is typical, the case

where the application acquires directly from the issuer presents some challenges such as connecting securely to a legitimate server.

[0070] In addition, there is value in making sure the application connects only to a legitimate address without the need to update the application for each new authorized issuer. In that case, the address to acquire a package from an issuer may include some digital signature or equivalent (e.g., message authentication code (MAC), crypto checksum, etc.). As such, the URI provided to the application, which has all the information to acquire a package, also embeds a digital signature that the application may use to verify and connect accordingly for starting the package acquisition. Typically, the URI includes a signature from a trusted authority known to the application. For example, in a preliminary step the issuer receives general information about the application including proper information to sign the URI for the application to initiate downloading.

[0071] Example of an issuer URI: [Issuer_Address]/[Package ID]/[Package Hash]/[Issuer Certificate].

[0072] The issuer certificate may be as short as the signed public key with a reference to the trusted authority. It is worth noting that the URI delivered by the issuer contains more information than the URI used to acquire a package.

Therefore, there is no problem with the typical 2083 character limit or some web services. The URI received by the application may contain much more information that does not have to be passed to the server side, such as:

[0073] (1) The signature and issuer certificate that are simply used for the application to verify that the received information is legitimate.

[0074] (2) The package identifier, which may be used in the payload of the HTTP request method used to acquire the package (e.g., the body of GET, PUT, POST, etc.).

[0075] (3) Additional security that may be put in place to ensure that only the target application is able to acquire the content. For example, secure delivery methods (e.g., secure channel to the application, separate channel to the end user, etc.) may be used for the URL. Additional authentication or identification information may be exchanged during application discovery and used to ensure the package is only opened by the target application instance. Additional steps may also be put in place to confirm this the user is authenticated, for example by using additional one-time code or other biometrics verification.

[0076] Finally, at the end of this process, the standalone package has been acquired. The issuer may have information to notify the application about package updates and the application may check for updates using information from the URI following a notification. Alternatively, the application may periodically (e.g. as per some schedule for the package) or regularly poll the issuer to determine whether any updates have occurred.

[0077] The URI of the package may be recorded by the application for later use. The URI to record or complementary information may also be delivered along with the package. The URI may be signed along the package data to ensure its integrity, authenticity and relevance to the package itself.

[0078] A package may be delivered along additional information to support the user experience:

[0079] (1) Information dedicated to the application UI such as:

[0080] (a) a label in end user language for each data element in a package

[0081] (b) a friendly name for the package, e.g., "Driving rights"

[0082] (c) a data type to help the user interface to properly display the information, e.g., string, date, URL, picture and other mime type, etc.

[0083] (2) Information for the UI for a verification:

[0084] (a) Naming a group of tags (not limited to the one from the delivered package) to simplify the presentation of the request where this name can be used as a purpose or as a mean to replace the group of tags instead of only showing the list of requested tag. Multiple groups of requested tags may share the same purpose. Having that information delivered by the issuer increases the trust level compared to receiving this in the request. It is also more convenient than having it delivered as part of the application code.

[0085] (3) Information for related enrollment:

[0086] (a) For example, a package may have references or information related to enrollment forms for additional credentials, aggregated or not. Such information may provide a mechanism to enroll for additional packages from within the application. The information may be delivered as a URI that contains the name of the package to enroll and a URI to the enroll site.

[0087] The use of the URI for these various features is also applicable to aggregated packages.

[0088] In the particular case of aggregated credentials, the issuance process may vary because of referenced data elements. One preliminary step may be to check if needed data elements are already issued and are valid to be referenced. In some embodiments, there is no need to access the value of such data elements and only the references are necessary. This provides value in regard to protecting personally identifying information (PII). Validity may simply mean issued by an adequate issuer. Checking if the needed data elements are already issued may be done either from within the application (e.g., when completing a form within the application) or a preliminary issuance step, which may involve the verification process or part of it. During that process an issuer may request for already issued data element(s) quite similarly to a request for verification. Already issued data elements may be delivered accordingly or authorization to fetch from online or a combination thereof.

[0089] In some embodiments, the request may be for references to data elements instead of the actual values of the data elements. This may be enabled by using specific tags for the references or having a parameter in the request to mean the list requested tag is to receive the references and not the actual value.

[0090] Information about the package containing the data element is also provided, including for example the URI of the package. After confirmation that a data element is valid to be referenced, the URI of the package may also be used as part of its referencing method as indicated above.

[0091] In some embodiments, it is only necessary to receive the information to reference the data element and not the data element itself. For instance, knowledge that such data exists from the adequate issuer is sufficient to be referenced. This may be the case, for example, when enrolling from the application itself where the application is able to perform the necessary checks to ensure valid data elements are present.

[0092] For data element missing from the response, the issuer may choose to enroll the end user for the missing data. In that case the issuer may use its own assurance level which might be lower than referencing from proper user. Alternatively, the issuer may direct the end user to enroll with an adequate issuer for such missing data.

[0093] Another difference with a standalone package is that an aggregated package would typically be made available to the same application whose data

elements would be referenced. Therefore, as the target application is typically known in advance, the issuance process of an aggregated package may be that the aggregated package is delivered or acquired automatically—without any specific user intervention in the acquisition process.

[0094] When an aggregated package is delivered or acquired automatically, the user may be Informed about the up-coming package and ask for consent before the package is prepared and not during the acquisition process. The user may also be requested to confirm that they are an authorized and correct user before acquisition begins. In some cases, the user may receive the package and then opt to delete it if not interested.

[0095] In some embodiments, the issuer of the aggregated package is provided with the necessary information to notify the source application or its gateway, which then triggers the acquisition of a package. The notification information may be delivered along the verification process, for example, using an additional command, a special tag to request such information as part of the request for verification command, or using similar application discovery as for standalone packages. In the case of using a special tag, a verification process may also be used to initiate the issuance of a standalone package. The application may, for example, return that information only to authorized identified verifiers.

[0096] Instead of notification, the issuer of the aggregated package may also provide the information to download the aggregated package along the with verification. For example, the issuer URI is provided along with or after or before the request.

[0097] Finally, once the package is ready, the target application or its gateway may be notified and the aggregated package is acquired without any specific action from the end user to trigger the acquisition. An issuer may have requested to be notified when data element(s) from one of its packages has been referenced. In such case, the package may have additional information to ensure the issuer is notified of such event. In that case, the application uses information from the package URI to notify the issuer. For example, a PUT, GET, etc. command is used to the specified URL and the body contains the package identifier along with information about the referencing issuer, e.g., the referencing package URL.

[0098] In some embodiments, the information about packages is kept on the backend (or the cloud) and even re-issuing referenced data to a device would notify the related issuer(s). While various embodiments may exist, one option is the backend to record both the URI of the referenced package and the URI(s) of the referencing issuer(s) so that an application receiving the same package may also receive information about referencing packages and the related issuers could be notified accordingly. Notifying the aggregated issuers can be done from the application or from the backend. The backend can be the application backend or a cloud backend where the user data is saved and can be restored or transferred. In some examples, the backup backend may only have references for all data elements and may not have any value thus not being exposed to PII.

[0099] FIG. 5 is a message sequence diagram illustrating electronic credential generation and issuance process 500, according to an embodiment. While the various processes will be described in accordance with illustrative steps performed in a particular order, it should be appreciated that embodiments of the present disclosure are not so limited and that any of the process steps depicted and described herein can be performed in any order or in parallel with one another.

[00100] A process is shown where a user requests an electronic credential via the mobile communication device 502 carried by the user. The user may be the person associated with the requested credential.

[00101] The process begins with issuers 504A or 504B transferring data to a credential database 506 (operation 550). For example, issuer 504A may be a financial institution and issuer 504B may be a governmental entity. In some embodiments, the data may be transferred over a secured communication session established over a communication network (e.g., a secured session using a hypertext transport protocol). The user of the mobile communication device 502 may initiate a request for an electronic credential (operation 552). For example, a person may request an insurance card in electronic format from a health insurance provider. The re-issuer 508, in this example a health insurance provider, requests data from the credential database 506 (operation 554), which may store data from issuers 504A and 504B. In some embodiments, re-issuer 506 may pay a fee in order to access the requested data. Once an appropriate payment has been received, credential database 506 transfers the requested data

to re-issuer 508 (operation 556). Such requested data could be only the references to needed data elements as long as the issuer meets the re-issuer criteria.

[00102] Re-issuer 508 generates the data to be issued (operation 558). In some embodiments, re-issuer 508 issues a standalone package that does not contain any referenced data. In other embodiments, the issued data includes at least one reference to data from another (one or more) package/issuer. The issuance process to the holder of mobile device 502 of a standalone package is different from issuing a package that contains referenced data because for a package that contains referenced data, the referenced data could be missing if the referenced package is gone and such data must be available at the time of verification. In some embodiments, packages containing referenced data may only be issued to applications where the reference data package is available. The re-issuer 508 issues the data package to mobile device 502 (operation 560).

[00103] In some embodiments, data packages may need to be updated. If the data package references data from other issuers, the referenced data may be updated. The re-issuer that referenced the data may desire to be notified of the update and verify the updated data before referencing the data again. For example, the re-issuer may be notified to fetch updated data. In addition, data packages may include information about the version of referenced data. In some embodiments, the re-issuer may not want to be notified as long as the same data tags exist and accept the updated data automatically. Additionally, the referencing data package should be alerted when the referenced data element/tag is removed or no longer present. In some embodiments, delivery of a data package with referenced data may trigger an update process to deliver the referenced package to the same application.

[00104] Using URI to identify package and reference data elements also provides additional benefits in regards to re-issuance. One aspect is to inform the issuer whose package data has been referenced and by which issuer(s). Accordingly, the issuer of the referenced package may notify the identified aggregated re-issuers when the same user gets the package reissued or a similar package issued. It also provides a mechanism to notify such re-issuers when similar packages are issued to other users.

[00105] Another aspect is for an end user to authorize another user to act on their behalf. For example, this may enable an end user to allow another person to use a pick-up receipt on their behalf. In that case, the package gets aggregated to another user identity. This may be done either by aggregating another identity to the existing one or by replacing the identity by the new one that the verifier checks. That process is initiated by the end user the package has been delivered to and may go through a backend system that re-issue for the new target user.

[00106] Another aspect is for an end user to receive a package for another user. For instance, this may be useful when multiple users have tickets from the same booking. In that case, it might be valuable for the user to also receive the information in an aggregated form so that all information may be presented at once. The aggregated information comes with the URI to acquire the missing referenced data. In that process authorization for the referenced data may have been delivered to the re-issuer so that the application from another user is able to acquire the information. For example, the referenced data application has delivered an authorization for the target end user or its application.

[00107] In some embodiments, verification may be implemented as a re-issuance instead of the typical process of responding to a request by tag. In that case the response may look like a package issued by (or on-behalf) of the holder application and aggregating information from the various packages whose requested granted data elements belong to.

Package Updates and Deletion

[00108] The update process may be similar to issuing aggregated credentials because the target application is known. Updating a package may be triggered by different events, such as notification from an issuer, scheduled pulling from the application, user triggered pulling, etc. The target package is acquired and the previous package with the same URI may be replaced by the new one. In some embodiments, the update process provides the hash of the current package. Then the application verifies that the acquired update references a hash that matches the current state of the package to update and only replaces the package accordingly. This is of interest in particular to prevent losing data that may have been modified through other mechanisms. For example, for a transactional data element that may be updated by another system and whose information has not

reconciled to the issuer backend. This is typical of transit tickets, which are delivered by the issuer and where a turnstile reader may modify the ticket data and the change isn't reported in real time to the issuer or not reported at all.

[00109] With aggregated credentials, updating a package may trigger additional events. In particular when an update may impact other packages because of referenced data elements.

[00110] While in some cases, an issuer may have chosen not to be informed as long as the referenced data element still exists, typical issuers may want to be notified when there is an update and the hash of the referenced data has changed or isn't available anymore. They may also want to be informed when there is an update and the hash of a package containing referenced data has changed, or when the package containing the referenced data has been deleted.

[00111] In some embodiments, the application notifies the related issuer using information from the URI of impacted package. This is done, for example, by using the HTTPS request method with information about the impacted package. Additional information about the referenced data elements may also be provided for information. A response to that notification may be a verification request for the issuer to analyze the data elements from the updated content.

[00112] Further events may also trigger a notification to an issuer. For example, when a package with a tag identical to one of the issuer packages is modified, it may be of interest for an issuer to know if a more legitimate issuer has issued information that may be referenced in the updated package. It may also be of interest to a competitor issuer.

[00113] Other information could be delivered along with a package including a digital signature to inform about authenticity and to allow verification of integrity. It is also a mean to bind the package information together. The signature typically applies to the package but other embodiments could rely on signing part of the package such as for example limiting the signature to the data element(s) and reference(s) to data elements.

[00114] Additionally, a reference to data element may be associated with the hash of the package. This provides a mechanism for an issuer to find out if the source package has been modified.

[00115] Access control information that may apply to all or specific data element(s) in a package may be delivered with the package. This provides a

mechanism to restrict access to specific data such as finger print to certain verifiers. Such access control mechanism may rely on a trusted authority where only a verifier whose identification certificate is issued by one of the trusted authorities is able to access the restricted information. Access control may also apply to the whole package.

[00116] Location information in the form for example of GPS coordinate(s) to determine a zone or beacon identifier may be delivered with the package. This provides a mechanism to wake up the application when such location condition is met. Typical use cases would be airline tickets or a pick-up receipt where the application is configured to notify the user about a verifier at a relevant time, e.g., when at or near the right gate or at or near the pick-up location.

[00117] Additionally, the information may be augmented with a pre-authorization for specific verifiers which would be pre-authorized to acquire specified data without the need for user consent.

[00118] Other information that may be included with the package include authorization to re-issue an aggregated package to another user or entity. For example, to transfer a pick-up receipt for another person to collect the good or service.

[00119] Additional information may include authorization to transfer the information to another application, e.g., share to another user or share to change application.

[00120] Additional information may include an expiration independent from the expiration date of the document. Such expiration may also be enforced by using tokenization.

[00121] An issued package may contain no data element or references. This could be used for example to deliver information for the application to register to wake up for specific events, locations or other conditions. Furthermore, temporary packages may be delivered empty except for the friendly name which may indicate a pending update or delivery. The actual name of the package may be updated along the acquisition of an update.

User Application

[00122] The user application executes on the user device and may access one or more issuers or re-issuers. The user application may also access the credential

database or other credential service. While various implementations of the user applications exist, they are typically implemented using one or more software components and have the capability to communicate with issuance system(s) and verification system(s).

[00123] A user application is capable to receive or acquire packages and may receive updates from one or more issuers. An application may also receive a request for verification containing a list of one or more tags each identifying a unique kind of data element and return a response. An application may have access to the data elements to obtain in the response or may provide an authorization to access such data elements from an issuer or a combination thereof.

[00124] User application user interfaces may include features or interfaces to enable Location Based Services (LBS). A user application or interface may optionally feature a notification interface.

[00125] A user application may be implemented in various programming languages or platforms, such as with an HTML 5 web page received from an issuer service or native application installed from a typical app store.

[00126] User applications may run on a variety of devices such as mobile phones, PCs, tablets, wearable devices, etc. Some implementations may also split the application across multiple devices. For example, the package acquisition agent may be implemented on a mobile phone while the verifier interface is implemented on a wearable device and the user interface may be present on both devices.

[00127] Some components of the application may also be implemented in a backend system instead of the user device. User applications may provide interfaces to different systems. Packages received may be stored in or out of the application itself as long as it is available to respond to a request.

[00128] Once a package is received, the user application keeps track of the download URL and uses this for checking for package updates. While the application may build the URI to record based on a received download URL and package identifier, a URI may be delivered with the package itself which could serve for several functions. For instance, the URI may provide a unique reference of the package or provide a URL to use for various commands.

[00129] The URL may be used to check for updates, e.g., HTTPS GET to download the package or an update. The URL may also be used to check for the status of the package or one of its particular a tag or a list of tags, e.g., HTTPs POST token ID to get a status. The URL may also be used to notify an issuer referencing information that an update has been acquired, e.g., HTTPS PUT to the referencing issuer with URI and hash of the referenced data that has changed. Additionally, the URL may be used to inform an issuer when a package has been deleted, for example, with an HTTPS DEL command, which may contain information about the application where the package was provisioned. In addition, the URL may be used to inform an issuer about an aggregation, for example, HTTPS PUT with information about the package with referenced data and the URI of the aggregated package.

[00130] The URI may also be used to reference data elements by adding the referenced tag to it and some information to ensure the matching data element is uniquely referenced such as the hash of the value of the referenced data element.

Verifier Operations

[00131] Verification is about requesting data element(s) by tag(s), then proceeding with the verification of the values (data elements). Additional information may be added to the verification request such as specifying a document type, a confidence level, terminal identification, etc.

[00132] Typically, the verifier sends a request containing the list of tags and receives a response. The response may include data elements matching some or all of the requested tags. Additional complementary information may also be delivered. The returned data is typically organized by “documents” as this is how the digital signature of the issuer applies. The verifier checks the validity of each document using, for example, an integrity check for the received data elements, authorized issuers, expected type of document, whether it meets a target level of confidence or assurance, etc.

[00133] In the case of aggregated credentials, a document may rely on information from different packages which may be from several issuers. As such, while the response format may be similar with packages returned in place of the documents, the part of the verification process where the data elements are

verified is slightly different and the UI may showcase the various issuers involved in a single document.

[00134] In some cases, the trust is implicit. For instance, if the verifier trusts the issuer of the aggregated package, then the assurance level of the referenced data is ignored and only the integrity is verified. In some other cases, the trust is explicit and then all issuers are verified as trusted.

[00135] There is also information that is present on many packages, such as expiration date. In some embodiments, such commonly used information may be returned automatically along with data element or referenced data element from the same package. In other embodiments, each have a unique tag yet such uniqueness may be built on a generic base for the kind of data and adding specific information related to the package or the kind of package, e.g., base is exp_date and exp_date_DL for expiration date of a driving privilege package thus avoiding creation of an all-new set of tags and allowing to reuse existing for the particular kind of package by adding the specific kind of “acronym”.

[00136] In some embodiments, in addition to the list of tag(s), the request includes a query (e.g., structured query language (SQL)) that aims at improving the user experience by facilitating the selection of data elements from different packages based on criteria defined in such a query. However, as the list of requested tags is provided separately, the SQL request is aimed to specify the expected relationship and a response may be provided even if some data from the SQL result set is missing. For example, the SQL query may be used to specify the expected value of a requested data element. This may be useful to request data elements for a local purpose such as pick up receipt or flight ticket. For example, the SQL query may be used to specify the relationship between different elements such as a car registration whose class of vehicle is matching with the corresponding tag from a driving license and where the card insurance is matching both the vehicle and the driver.

[00137] The SQL query may also be used to quantify information not in the data elements such as indicating a minimum level of confidence (assurance) or specify a given issuer.

[00138] From a user experience point of view, the SQL query may be used to drive the various steps for a user to select adequate packages for the response. Following the SQL query, packages that are not adequate are filtered out then

the filtering process continues following the selection of first data elements or package reducing further the choices based on pre-selection. For example, the SQL query may specify that tags for vehicle category from car registration should match with the vehicle category from a driving license then that car insurance should match both car registration and identity information the driving license is aggregated to.

[00139] In some embodiments, the request may be received as different steps where one step is a broadcast which could be generic or targeted and implemented as multiple steps, e.g., beacon to wake up the application (e.g., based on location-based services (LBS) information that has been delivered with a package), then application acquires more details. Another step may follow where the application may notify the user and communicate with the verifier to receive all of the remaining details of the request using the same process as for a typical verification. For example, the first step may be for the request to notify a driver of a car by broadcasting the information about the plate number. Then, applications that do not have data matching that plate number information ignore the broadcast, whereas other application(s) may connect to receive the second part of the request and notify the user or connect accordingly for the continuing with the request/response process or just for sending the response.

[00140] For these implementations, a first part of the request may contain information for the application to decide if the request should be ignored or not, information about the notification information and about the call for action (e.g. for the button to click). Then, the application may connect to get the second part of the request which may bring the specific tags and relationship information. Alternatively, the application may connect after the call for action. As such, depending on the process, the notification may close and the process continues on the verifier side. Alternatively, the notification opens the application where the second part of the request may be received and where the user may have more control on what is being shared. For example, the user may provide confirmation with a personal identification number (PIN) or biometrics or view in more details the content of the request or what is going to be returned.

[00141] A user interface (FIG. 6) may be presented to prompt the user for access control to the application. Then, the user may provide consent to share specific information based on the information that matches the request (FIG. 7).

[00142] Once the data elements have been received and the issuer identified, some or all of the information may be recorded in a blockchain while both the verifier information may remain anonymous as well as the end user. This permits the issuer to verify the number of transactions and claim a revenue share from the cumulated transactions. In that particular case, the information that needs to be reported is the URI of the package whose data element has been verified. There is no need to report the verifier or information about the end user information. That being said, in some embodiments where the application checks the blockchain for verification, information about the verifier may be reported as well.

[00143] There are also special cases where multiple packages have tags that match partially or entirely to those in the request. All of the partially or fully matched tags may be returned at once. An example use case is for flight tickets. The boarding gate sends a request for particular flight ticket data and the response comes back with data for multiple tickets.

[00144] Taking advantage of the URI and notification, an end user may be notified every time his or her information is presented to a verifier. The URI associated with the package provides a mechanism for the issuer to be notified and such issuer may use the notification mechanisms to inform the end user through the application. Such process enhances the overall security and provides more confidence in the system.

Operations

[00145] FIG. 8 is a flowchart illustrating a method 800 of installing and registering an aggregated credential user application, according to an embodiment. The method 800 begins at operation 802, when the aggregated credential user application and data packages associated therewith are respectively downloaded and installed onto a mobile device.

[00146] At 804, the method 800 continues securely requesting an electronic credential. In embodiments, the request may indicate a minimum level of assurance or specific data tags. User identity may be verified before the request for the electronic credential may be transferred. For example, the user may use fingerprint verification or enter a password. In some embodiments, the verification may comprise a two-step authentication.

[00147] At 806, the method 800 then continues with generating the issued data, which may include a standalone data package or a data package that references data from other issuers. For example, the aggregate credential system responds with one or more data element(s) for the requested data tags. In other words, various data packages may match the requested data. In some embodiments, the requestor may specifically request only one data element be returned and may additionally specify the issuer. In embodiments, a response may include data elements with the same data tag from multiple packages. For example, in a request for last name and a boarding pass, the response may include boarding passes for each member of a family traveling together. In another example, a request for first and last name and car insurance cards may return all car insurance policies associated with an individual.

[00148] The aggregated credential user application issues the electronic credential/data package to the mobile communication device (operation 808). The re-issuer may present the issued data grouped by issuer. The data may be referenced from one or more other issuers (or re-issuers) and the assurance level may be associated or linked to issuer, providing the necessary data to present the sorted or grouped user interface. Grouping by issuers facilitates the verification of the digital signature of the issued data. Aggregating credentials allows data to be packaged or grouped by issuers and combining new data elements with referenced data elements.

Example Use Case

[00149] In a preliminary step, the end user is informed about an application(s) to use for the credentials of a given issuer. Installation of the application may be driven from a list of authorized applications from the given issuer, which may also embed links to install the application from an adequate app store.

[00150] Once an application has been installed, the end user may request a standalone package from within the application or otherwise information to trigger the acquisition is redeemed in the application. In the case where the request is performed from within the application and the end user opts for the package to be delivered to that same application, the package is shown as pending when the request has been submitted and then changes after the actual package has been received. The package could show empty until confirmed or

better it shows with all data elements yet marked as pending. Alternatively, the application may acquire a standalone package by receiving the address to connect to and download the target package using common techniques such as MagicLink or DeepLinking, etc. where the URL contains information for both the target application and the address of the target package.

[00151] While the target package may be identified solely by a unique identifier when the application knows which backend to connect to, in the more generic implementation the full URL to acquire the target package is received. For example, the URI for a package

“https://issuer_Address.com/package_identifier/hash/signature/etc” may be encoded as a parameter to append to the URL to call the application as follows: “https://issuer_Address.com/package_identifier/hash/signature/etc”. Finally, a digital signature that has been added to the parameters is checked to ensure the application will not connect to a malicious site or address.

[00152] To authorize an application a connection uses TLS with mutual authentication and a sole application is authorized for the package. In such a case, the application certificate is received (for example during the identification of the application that could be used by an issuer to determine if the application is legitimate), then pinned by the issuer to control access to the content. For the TLS session the issuer site has a certificate from an authorized trusted authority (TA) of the application.

[00153] As an alternative, to authorize the application other security techniques such as chaining, combining information provided by the phone browser such as model, etc., may be used.

[00154] More standalone packages may be acquired following the available process. All packages are delivered with a URI for the application to check for updates and for data elements to be referenced. The packages may also be delivered with the necessary data to fetch registration templates for additional packages – typically aggregated – that may be requested from within the application. Such registration templates may embed information that makes it easy for the user interface to pick data element(s) from package(s) that should be referenced

[00155] In some embodiment, the user interface may only provide the option to register the new information and take advantage of the template information

to automatically select appropriate data element(s) to reference from package(s) where it belongs. The template information may specify the tag of the data elements to reference and either or a combination of for example the kind of document, the issuer, the name of the package, or other information that allows to identify a package without using personalized data such as a package identifier.

[00156] To acquire an aggregated package the user may register from within the application for a document that results in one or more aggregated packages. A verification to acquire information about the data elements to reference and more particularly the URI of the package is then used. The application receives the URI of the aggregated package during these preliminary steps and the user interface shows a pending package as well as related information to the package. In the case of registering from within the application, the pending package may show all the information that has been submitted and only inform that it is pending (for example because the issuer signature has not been received yet).

[00157] After the package is downloaded, the update process kicks in (notification to the application from the issuer, scheduled pulling, etc.) and the package is updated. In some cases, the issuer (or re-issuer) delivers a package that is able to overwrite the existing one and in some other cases the issuer delivers only the information that has been modified or added. The package is signed for integrity, authenticity, and for binding the data elements and referenced data element together.

[00158] Once a package URI is received, it may refer to a new package or an update to an existing package. The application uses an HTTPS GET commands with information from the URI to acquire the package. The body of the command may contain the package identifier and other information from the URI. Once the package is received and has been checked for integrity, the application may execute the applicable rules from the various issuers involved (i.e., including rules from issuer from referenced package) and may fetch the missing packages, notify the adequate issuer referencing the package about the update, notify the adequate referenced package issuer(s) about the referencing issuer, or set a schedule for checking for updates for that particular package.

[00159] The application uses an HTTPS PUT command with information from the URI to notify an issuer about the event related to their issued package. The

body in the PUT command contains information about the kind of event and provides the URI of the other issuer package.

[00160] Overall the application simply manages URI to check for updates and to maintain the relationship between packages. Updates are triggered automatically from notification from the individual issuers.

[00161] The URI of acquired package(s) is also recorded in a backend system where the information is protected to ensure only the right user can authorize access to the information. Such back up facilitates changing mobile devices and when authorized to install the packages on multiple devices owned by the same user.

[00162] In some cases, the PUT commands are also used to interface with a blockchain that is personal to the end user and the information is populated either by the end user application or by the issuers or both.

[00163] When a package is removed from the application, the application notifies the issuer with an HTTPS DEL command and using information from the deleted package URI. This may also be used to confirm that a package has been deleted from the application, the backend or marked as deleted in the personal blockchain.

[00164] Issuers of related packages may be notified about the event as well using an HTTPS PUT command as described above.

[00165] For verification, the end user engages with a verifier and the application receives the request. The request also embeds a query which drives the steps in the user interface and makes it easy for the end user to select and consent to share the requested data elements. The verifier receives the response that may contain multiple packages. In the case where a package has been tokenized, the verifier may use the HTTPS POST command with information from the package URI to check about the status of the current token for the package.

Example System Components

[00166] Embodiments may be implemented in one or a combination of hardware, firmware, and software. Embodiments may also be implemented as instructions stored on a machine-readable storage device, which may be read and executed by at least one processor to perform the operations described herein. A

machine-readable storage device may include any non-transitory mechanism for storing information in a form readable by a machine (e.g., a computer). For example, a machine-readable storage device may include read-only memory (ROM), random-access memory (RAM), magnetic disk storage media, optical storage media, flash-memory devices, and other storage devices and media.

[00167] A processor subsystem may be used to execute the instruction on the machine-readable medium. The processor subsystem may include one or more processors, each with one or more cores. Additionally, the processor subsystem may be disposed on one or more physical devices. The processor subsystem may include one or more specialized processors, such as a graphics processing unit (GPU), a digital signal processor (DSP), a field programmable gate array (FPGA), or a fixed function processor.

[00168] Examples, as described herein, may include, or may operate on, logic or a number of components, modules, or mechanisms. Modules may be hardware, software, or firmware communicatively coupled to one or more processors in order to carry out the operations described herein. Modules may be hardware modules, and as such modules may be considered tangible entities capable of performing specified operations and may be configured or arranged in a certain manner. In an example, circuits may be arranged (e.g., internally or with respect to external entities such as other circuits) in a specified manner as a module. In an example, the whole or part of one or more computer systems (e.g., a standalone, client or server computer system) or one or more hardware processors may be configured by firmware or software (e.g., instructions, an application portion, or an application) as a module that operates to perform specified operations. In an example, the software may reside on a machine-readable medium. In an example, the software, when executed by the underlying hardware of the module, causes the hardware to perform the specified operations. Accordingly, the term hardware module is understood to encompass a tangible entity, be that an entity that is physically constructed, specifically configured (e.g., hardwired), or temporarily (e.g., transitorily) configured (e.g., programmed) to operate in a specified manner or to perform part or all of any operation described herein. Considering examples in which modules are temporarily configured, each of the modules need not be instantiated at any one moment in time. For example, where the modules comprise a general-purpose

hardware processor configured using software; the general-purpose hardware processor may be configured as respective different modules at different times. Software may accordingly configure a hardware processor, for example, to constitute a particular module at one instance of time and to constitute a different module at a different instance of time. Modules may also be software or firmware modules, which operate to perform the methodologies described herein.

[00169] Circuitry or circuits, as used in this document, may comprise, for example, singly or in any combination, hardwired circuitry, programmable circuitry such as computer processors comprising one or more individual instruction processing cores, state machine circuitry, and/or firmware that stores instructions executed by programmable circuitry. The circuits, circuitry, or modules may, collectively or individually, be embodied as circuitry that forms part of a larger system, for example, an integrated circuit (IC), system on-chip (SoC), desktop computers, laptop computers, tablet computers, servers, smart phones, etc.

[00170] As used in any embodiment herein, the term “logic” may refer to firmware and/or circuitry configured to perform any of the aforementioned operations. Firmware may be embodied as code, instructions or instruction sets and/or data that are hard-coded (e.g., nonvolatile) in memory devices and/or circuitry.

[00171] “Circuitry,” as used in any embodiment herein, may comprise, for example, singly or in any combination, hardwired circuitry, programmable circuitry, state machine circuitry, logic and/or firmware that stores instructions executed by programmable circuitry. The circuitry may be embodied as an integrated circuit, such as an integrated circuit chip. In some embodiments, the circuitry may be formed, at least in part, by the processor circuitry executing code and/or instructions sets (e.g., software, firmware, etc.) corresponding to the functionality described herein, thus transforming a general-purpose processor into a specific-purpose processing environment to perform one or more of the operations described herein. In some embodiments, the processor circuitry may be embodied as a stand-alone integrated circuit or may be incorporated as one of several components on an integrated circuit. In some embodiments, the various

components and circuitry of the node or other systems may be combined in a system-on-a-chip (SoC) architecture

[00172] FIG. 9 is a block diagram illustrating a machine in the example form of a computer system 900, within which a set or sequence of instructions may be executed to cause the machine to perform any one of the methodologies discussed herein, according to an embodiment. In alternative embodiments, the machine operates as a standalone device or may be connected (e.g., networked) to other machines. In a networked deployment, the machine may operate in the capacity of either a server or a client machine in server-client network environments, or it may act as a peer machine in peer-to-peer (or distributed) network environments. The machine may be a mobile device, a personal computer (PC), a tablet PC, a hybrid tablet, a personal digital assistant (PDA), a mobile telephone, or any machine capable of executing instructions (sequential or otherwise) that specify actions to be taken by that machine. Further, while only a single machine is illustrated, the term “machine” shall also be taken to include any collection of machines that individually or jointly execute a set (or multiple sets) of instructions to perform any one or more of the methodologies discussed herein. Similarly, the term “processor-based system” shall be taken to include any set of one or more machines that are controlled by or operated by a processor (e.g., a computer) to individually or jointly execute instructions to perform any one or more of the methodologies discussed herein.

[00173] Example computer system 900 includes at least one processor 902 (e.g., a central processing unit (CPU), a graphics processing unit (GPU) or both, processor cores, compute nodes, etc.), a main memory 904 and a static memory 906, which communicate with each other via a link 908 (e.g., bus). The computer system 900 may further include a video display unit 910, an alphanumeric input device 912 (e.g., a keyboard), and a user interface (UI) navigation device 914 (e.g., a mouse). In one embodiment, the video display unit 910, input device 912 and UI navigation device 914 are incorporated into a touch screen display. The computer system 900 may additionally include a storage device 916 (e.g., a drive unit), a signal generation device 918 (e.g., a speaker), a network interface device 920, and one or more sensors (not shown), such as a global positioning system (GPS) sensor, compass, accelerometer, gyrometer, magnetometer, or another sensor.

[00174] The storage device 916 includes a machine-readable medium 922 on which is stored one or more sets of data structures and instructions 924 (e.g., software) embodying or utilized by any one or more of the methodologies or functions described herein. The instructions 924 may also reside, completely or at least partially, within the main memory 904, static memory 906, and/or within the processor 902 during execution thereof by the computer system 900, with the main memory 904, static memory 906, and the processor 902 also constituting machine-readable media.

[00175] While the machine-readable medium 922 is illustrated in an example embodiment to be a single medium, the term “machine-readable medium” may include a single medium or multiple media (e.g., a centralized or distributed database, and/or associated caches and servers) that store the one or more instructions 924. The term “machine-readable medium” shall also be taken to include any tangible medium that is capable of storing, encoding or carrying instructions for execution by the machine and that cause the machine to perform any one or more of the methodologies of the present disclosure or that is capable of storing, encoding or carrying data structures utilized by or associated with such instructions. The term “machine-readable medium” shall accordingly be taken to include, but not be limited to, solid-state memories, and optical and magnetic media. Specific examples of machine-readable media include non-volatile memory, including but not limited to, by way of example, semiconductor memory devices (e.g., electrically programmable read-only memory (EPROM), electrically erasable programmable read-only memory (EEPROM)) and flash memory devices; magnetic disks such as internal hard disks and removable disks; magneto-optical disks; and CD-ROM and DVD-ROM disks.

[00176] The instructions 924 may further be transmitted or received over a communications network 926 using a transmission medium via the network interface device 920 utilizing any one of a number of well-known transfer protocols (e.g., HTTP). Examples of communication networks include a local area network (LAN), a wide area network (WAN), the Internet, mobile telephone networks, plain old telephone (POTS) networks, and wireless data networks (e.g., Bluetooth, Wi-Fi, 3G, and 4G LTE/LTE-A, 5G, DSRC, or WiMAX networks). The term “transmission medium” shall be taken to include

any intangible medium that is capable of storing, encoding, or carrying instructions for execution by the machine, and includes digital or analog communications signals or other intangible medium to facilitate communication of such software.

Additional Notes:

[00177] The above detailed description includes references to the accompanying drawings, which form a part of the detailed description. The drawings show, by way of illustration, specific embodiments that may be practiced. These embodiments are also referred to herein as “examples.” Such examples may include elements in addition to those shown or described. However, also contemplated are examples that include the elements shown or described. Moreover, also contemplated are examples using any combination or permutation of those elements shown or described (or one or more aspects thereof), either with respect to a particular example (or one or more aspects thereof), or with respect to other examples (or one or more aspects thereof) shown or described herein.

[00178] Publications, patents, and patent documents referred to in this document are incorporated by reference herein in their entirety, as though individually incorporated by reference. In the event of inconsistent usages between this document and those documents so incorporated by reference, the usage in the incorporated reference(s) are supplementary to that of this document; for irreconcilable inconsistencies, the usage in this document controls.

[00179] In this document, the terms “a” or “an” are used, as is common in patent documents, to include one or more than one, independent of any other instances or usages of “at least one” or “one or more.” In this document, the term “or” is used to refer to a nonexclusive or, such that “A or B” includes “A but not B,” “B but not A,” and “A and B,” unless otherwise indicated. In the appended claims, the terms “including” and “in which” are used as the plain-English equivalents of the respective terms “comprising” and “wherein.” Also, in the following claims, the terms “including” and “comprising” are open-ended, that is, a system, device, article, or process that includes elements in addition to those listed after such a term in a claim are still deemed to fall within the scope of that

claim. Moreover, in the following claims, the terms “first,” “second,” and “third,” etc. are used merely as labels, and are not intended to suggest a numerical order for their objects.

[00180] The above description is intended to be illustrative, and not restrictive. For example, the above-described examples (or one or more aspects thereof) may be used in combination with others. Other embodiments may be used, such as by one of ordinary skill in the art upon reviewing the above description. The Abstract is to allow the reader to quickly ascertain the nature of the technical disclosure. It is submitted with the understanding that it will not be used to interpret or limit the scope or meaning of the claims. Also, in the above Detailed Description, various features may be grouped together to streamline the disclosure. However, the claims may not set forth every feature disclosed herein as embodiments may feature a subset of said features. Further, embodiments may include fewer features than those disclosed in a particular example. Thus, the following claims are hereby incorporated into the Detailed Description, with a claim standing on its own as a separate embodiment. The scope of the embodiments disclosed herein is to be determined with reference to the appended claims, along with the full scope of equivalents to which such claims are entitled.

ABSTRACT

Various systems and methods for managing aggregated credentials are described herein. In an example, the system is configured to generate and issue electronic credentials that may include aggregate credentials from various issuers.

Acquiring a standalone package

Once an adequate application has been installed, the end user may request a standalone package from within the application or information to trigger the acquisition is redeemed in the application.

In the case where the request is done from within the application and the end user opts for the package to be delivered to that same App, the package is shown as pending when the request has been submitted then changes after the actual package has been received.

Alternatively, the application acquires a standalone package by receiving the address to connect to download the target package using common techniques such as MagicLink or DeepLinking, etc. whereas the URL contain information for both the target application and the address of the target package:

- While the target package can be identified solely by a unique identifier when the application knows which backend to connect to, in the more generic implementation the full URL to acquire the target package is received. For example the URI for a package `https://issuer_Address.com/package_identifier/hash/signature/etc` can be encoded as a parameter to append to the URL to call the application as follows: `https-issuer_Address.com/package_identifier/hash/signature/etc`.
- Finally, a digital signature that has been added to the parameters is check to ensure the application would not connect to a malicious site/address then connection is decided accordingly.

Different techniques could apply to authorize an application:

- Connection uses TLS with mutual authentication and a sole application is authorized for the package: in that case the application certificate is received (for example during the identification of the application that could be used by an issuer to determine if the application is legitimate) then pinned by issuer to control access to the content. For the TLS session the issuer site has a certificate from an authorized trusted authority (TA) of the application.
- or other security techniques such as: chaining, combining information provided by the phone browser such as model, etc.

More standalone package can be acquired following the available process.

In that optimum solution, all packages are delivered with a URI for the App to check for updates and for data element to be referenced. The packages may also be delivered with the necessary data to fetch registration template for additional packages – typically aggregated – that could be requested from within the app. Such registration template could embed information that makes it easy for the user interface to pick data element(s) from package(s) that should be referenced. In some embodiment, the UI may only provide the option to register the new information and take advantage of the template information to select adequate data element(s) to reference from matching package(s). The template information could specify the tag of the data elements to reference and either or a combination of for example the kind of document, the issuer, the name of the package, or other information that allows to identify a package without using personalized data such as a package identifier.

Acquiring an aggregated package

Preliminary steps to the acquisition could be

- Registering from within the application for a document that results in one or more aggregated packages
- Involving a verification to acquire information about the data elements to reference and more particularly the URI of the package

In that optimum approach, it is envisioned that the application receives the URI of the aggregated package during these preliminary steps and the UI shows a pending package as well as related information to the package. In the case of registering from within the application, the pending package could show all the information that has been submitted and only inform that it is pending (for example because the issuer signature has not been received yet).

Then update process kicks in (notification to the application from the issuer, scheduled pulling, etc.) and the package is updated. In some cases, the issuer delivers a package that can overwrite the existing one and in some other cases the issuer delivers only the information that has been modified and that is added. More importantly, in that optimum approach, the package is signed for integrity, authenticity and for binding the data elements and referenced data element together.

Package installation & update

Once a package URI is received, it may be new or an update.

In that optimum implementation example, the application uses HTTPs GET command with information from the URI to acquire the package and whereas the body may contain the package identifier and other information from the URI

Once the package is received and has been checked for integrity, the application may execute the applicable rules from the various issuers involved (i.e. including rules from issuer from referenced package) and may:

- fetch the missing packages
- notify the adequate issuer referencing the package about the update
- notify the adequate referenced package issuer(s) about the referencing issuer
- set a schedule for checking for updates for that particular package

In that optimum implementation example, the application uses HTTPs PUT command with information from the URI to notify an issuer about the event related to their issued package and whereas the body contains information about the kind of event and provides the URI of the other issuer package.

Overall the application simply manages URI to check for updates and to maintain the relationship between packages.

Updates are triggered automatically from notification from the individual issuers.

In this optimum implementation, the URI of acquired package(s) is also recorded in a backend system where the information is protected to ensure only the right user can authorize access to the information.

Such back up facilitates changing mobile devices and when authorized to install the packages on multiple devices owned by the same user.

In some cases, the PUT commands are also used to interface with a blockchain personal to the end user and the information is populated either by the end user application or by the issuers or both.

Deleting a package

When a package removed from the application, the optimum application notifies the issuer with HTTPs DEL command and using information from the deleted package URI. This could also be used to confirm that a package has been deleted from the application, the backend or marked as deleted in the personal blockchain.

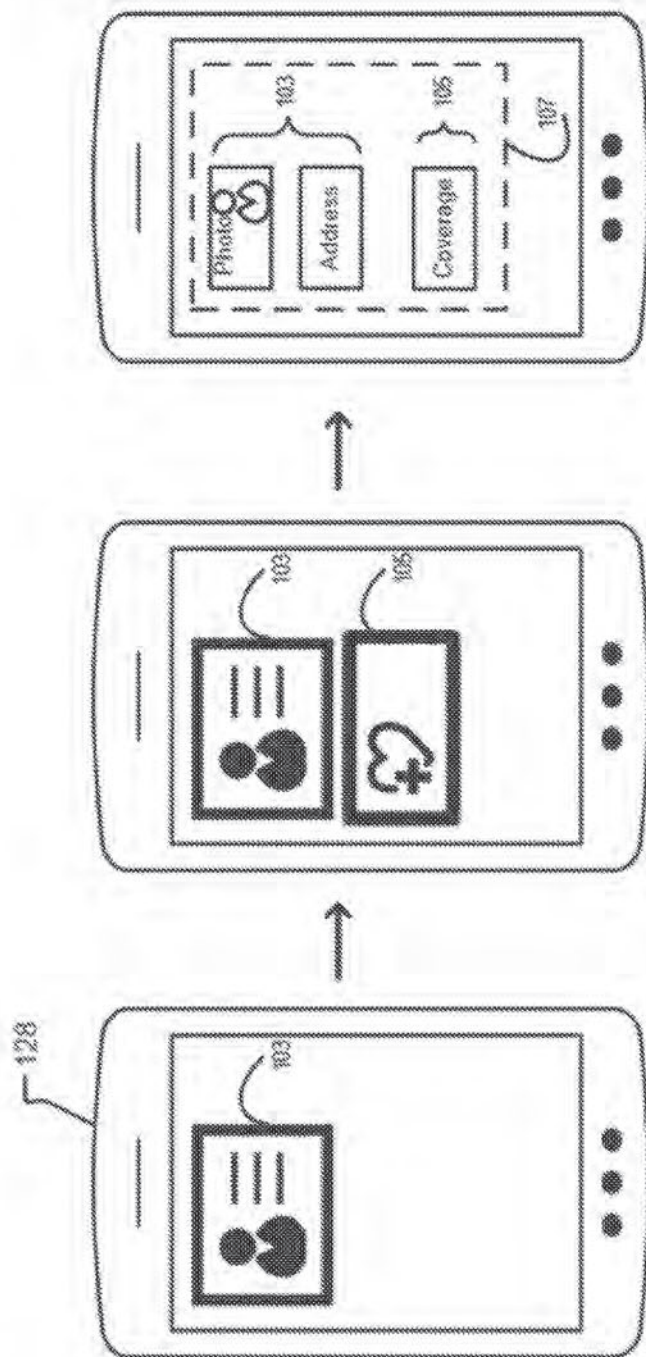
Issuers of related packages can be notified about the event as well using HTTPs PUT command as described above.

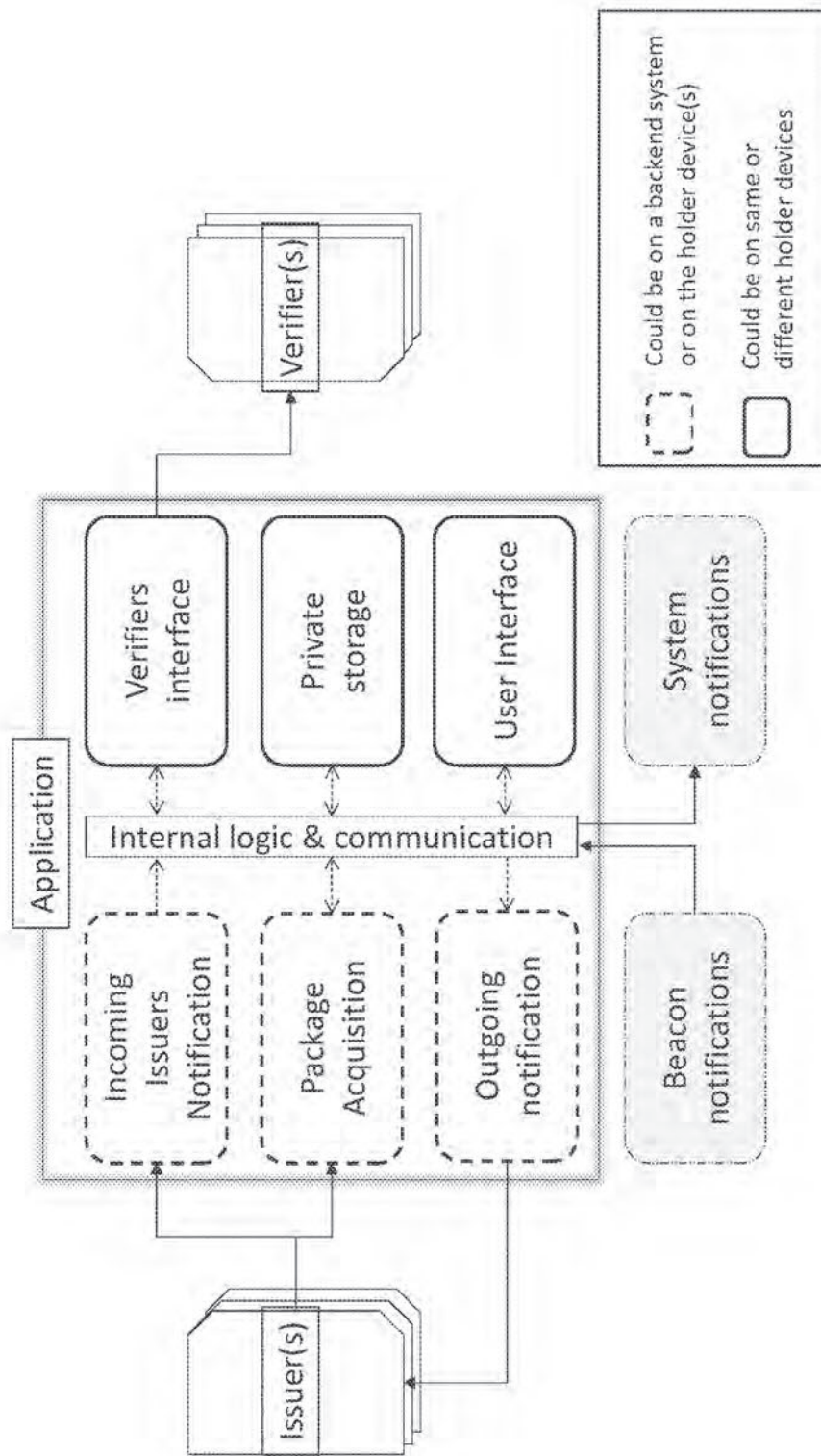
Verification

The end user engages with a verifier and the application receives the request.

In that optimum implementation, the request also embeds a query which drives the steps in the user interface and makes it easy for the end user to select and consent to share the requested data elements.

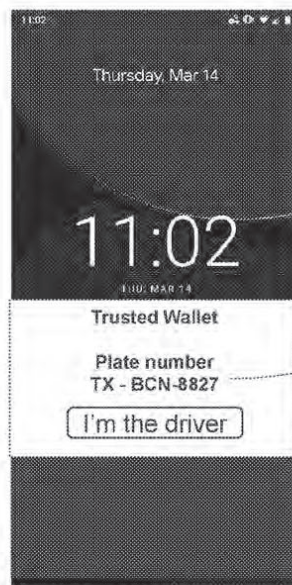
The verifier receives the response that may contain multiple packages. In the case where a package has been tokenized, the verifier could use the HTTPs POST command with information from the package URI to check about the status of the current token for the package.







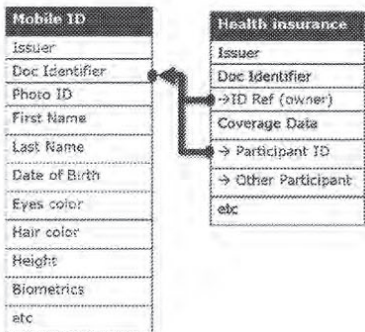
Generic broadcast
Resulting in a user notification



Targeted broadcast
resulting in a notification
to matching user



SQL Pre-selected data



- Some packages can include and/or reference a given Tag multiple times
 - It could be identical Tag from different packages for example different people associated on the same package for an insurance contract
 - The same data could be references multiple times in the same package e.g. the insurance contract representative/owner is also a participant covered by the insurance
 - A package could also reference a Tag from its own package

Passport attributes

Utopian Republic

First name: [text field]

Last name: [text field]

Address: [text field]

Utopian Government

Citizenship: [text field]

Date expiration du passeport: [text field]

SEND

Example of UI for a form referencing data from another package

Passport Attribute

Utopian Republic

First name: [value]

Last name: [value]

Address: [value]

Utopian Government

Citizenship: [value]

Date expiration du passeport: [value]

Example of UI showing the name of packages to the holder

Name of request

Utopian Republic

First name: [value]

Last name: [value]

Address: [value]

Utopian Government

Citizenship: [value]

Date expiration du passeport: [value]

First Name: [value]

Family Name: [value]

Home Address: [value]

Example of UI showing the name of the issuer on the verifier

SYSTEM AND METHOD FOR PROVIDING AGGREGATED CREDENTIALS WITH ASSURANCE LEVELS

FIELD OF THE DISCLOSURE

[0001] Embodiments of the present disclosure relate to an aggregated credential data structure and, more specifically, to aggregating credentials, verifying credentials, and issuing credentials.

BACKGROUND

[0002] Aggregating user credentials and/or data from one or more credential issuers, to provide a newly issued document with verified data/credentials. Many entities store user data and issue credentials, such as the government, Department of Motor Vehicles (DMV), universities, healthcare providers, financial institutions, utilities companies, airlines, etc. These entities may store similar and/or unique user data. For instance, a government issued identification card or driver's license may include the associated individual's picture, name, date of birth (DOB), height, weight, eye color, social security number (SSN), etc. An individual's student ID may include some of the same information, but may also include a student ID number that is unique to the school issuing the ID. Additionally, each of these identification documents (e.g., driver's license and student ID) may have an issue data and/or an expiration date. It may be cumbersome for a person to have multiple documents (e.g., driver's license, health insurance card, student ID, etc.) in his wallet all. Furthermore, the documents may be easily lost, misplaced, or stolen. Many people often carry a mobile communication device and would prefer to have their documents accessible via their mobile communication device rather than carrying separate physical documents. While user experience is improved, the problems associated with aggregating the data and counterfeit or duplicated electronic documents still need to be addressed.

SUMMARY

[0003] It is with respect to the above that embodiments of the present disclosure were contemplated. In particular, embodiments of the present disclosure contemplate various systems and methods. One non-limiting example of a system includes:

[0004] A system comprising:

- a data aggregation server;
- a communication interface coupled with the processor and facilitating machine-to-machine communications over a communications network; and
- a computer-readable storage medium coupled with the processor, wherein the computer-readable storage medium comprises instructions that, when executed by the processor, enable the processor to:
 - request and receive data from one or more issuers for an individual, wherein the data is identified with a unique identifier, and wherein each data element is associated with a specified tag;
 - verify and store the data, wherein each data element is stored with a source tag indicating the source; and
 - prepare and present data package to issue in response to a data request.

[0005] A system comprising:

- a data aggregation server;
- a communication interface coupled with the processor and facilitating machine-to-machine communications over a communications network; and
- a computer-readable storage medium coupled with the processor, wherein the computer-readable storage medium comprises instructions that, when executed by the processor, enable the processor to:
 - receive a request from a requesting issuer for data from one or more other issuers;
 - generating a data package, wherein the data package comprises referenced data from the one or more other issuers and issued data from the requesting issuer; and
 - issuing the data package, wherein each data element in the data package includes an assurance level.

[0006] A method comprising:

- receiving a request for an electronic credential from an issuer, wherein the credential references at least one data element from one or more other issuers;
- generating the electronic credential; and
- issuing the electronic credential to a mobile device.

[0007] The phrases one or more of “at least one,” “one or more,” and “and/or” are open-ended expressions that are both conjunctive and disjunctive in operation. For example, each of the

expressions “at least one of A, B, and C,” “one or more of A, B, or C” and “A, B, and/or C” means A alone, B alone, C alone, A and B together, A and C together, B and C together, or A, B and C together. When each one of A, B, and C in the above expressions refers to an element, such as X, Y, and Z, or class of elements, such as X1-Xn, Y1-Ym, and Z1-Zo, the phrase is intended to refer to a single element selected from X, Y, and Z, a combination of elements selected from the same class (e.g., X1 and X2) as well as a combination of elements selected from two or more classes (e.g., Y1 and Zo).

[0008] The term “a” or “an” entity may refer to one or more of that entity. As such, the terms “a” (or “an”), “one or more” and “at least one” can be used interchangeably herein. It is also to be noted that the terms “comprising”, “including”, and “having” can be used interchangeably.

[0009] The preceding is a simplified summary of the disclosure to provide an understanding of some aspects of the disclosure. This summary is neither an extensive nor exhaustive overview of the disclosure and its various aspects, embodiments, and configurations. It is intended neither to identify key or critical elements of the disclosure nor to delineate the scope of the disclosure but to present selected concepts of the disclosure in a simplified form as an introduction to the more detailed description presented below. As will be appreciated, other aspects, embodiments, and configurations of the disclosure are possible utilizing, alone or in combination, one or more of the features set forth above or described in detail below.

BRIEF DESCRIPTION OF THE DRAWINGS

[0010] The following description and associated figures teach the best mode of the invention. For the purpose of teaching inventive principles, some conventional aspects of the best mode may be simplified or omitted. The following claims specify the scope of the invention. Note that some aspects of the best mode may not fall within the scope of the invention as specified by the claims. Thus, those skilled in the art will appreciate variations from the best mode that fall within the scope of the invention. Those skilled in the art will appreciate that the features described below can be combined in various ways to form multiple variations of the invention. As a result, the invention is not limited to the specific examples described below, but only by the claims and their equivalents.

[0011] The present disclosure is described in conjunction with the appended figures, which are not necessarily drawn to scale:

[0012] Fig. 1 is a block diagram depicting an aggregated credential system in accordance with at least some embodiments of the present disclosure;

[0013] Fig. 2 is a block diagram depicting a data structure for receiving and storing aggregated credentials in accordance with at least some embodiments of the present disclosure;

[0014] Fig. 3 is a depicting an aggregated credential system in accordance with at least some embodiments of the present disclosure;

[0015] Fig. 4 is a flow diagram depicting the operation of the aggregated credential system in accordance with at least some embodiments of the present disclosure;

[0016] Fig. 5 is a block diagram depicting details of a mobile communication device in accordance with at least some embodiments of the present disclosure;

[0017] Fig. 6 is a block diagram depicting details of an aggregated credential server in accordance with at least some embodiments of the present disclosure;

[0018] Fig. 7 is a diagram depicting details of the aggregated credential service providing location-based services in accordance with at least some embodiments of the present disclosure;

[0019] Fig. 8 is a flow diagram depicting a method of generating and issuing an electronic credential in accordance with at least some embodiments of the present disclosure.

DETAILED DESCRIPTION

[0020] COPYRIGHT AND LEGAL NOTICES

A portion of the disclosure of this patent document includes material which is subject to copyright protection. The copyright owner has no objection to the facsimile reproduction by anyone of the patent document or the patent disclosure, as it appears in the Patent and Trademark Office patent files or records, but otherwise reserves all copyrights whatsoever.

[0021] Before any embodiments of the disclosure are explained in detail, it is to be understood that the disclosure is not limited in its application to the details of construction and the arrangement of components set forth in the following description or illustrated in the following drawings. The disclosure is capable of other embodiments and of being practiced or of being carried out in various ways. Also, it is to be understood that the phraseology and terminology used herein is for the purpose of description and should not be regarded as limiting.

[0022] In some embodiments, an issuer may receive full or a subset of data for one or more other issuers. The data may be electronic format that includes a unique identifier of the source

document. The issuer may provide new data to be issued. The aggregated credential system prepares the data to be issued. In some examples, the data to be issued references data received from the one or more other issuers. The aggregated credentials system issues a new document that may reference data received from the one or more other issuers and includes new data from the issuer. In some embodiments, the new data may be signed by the issuer.

[0023] With reference initially to Fig. 1, a data issuer may request data from another issuer. In some embodiments, the request may specify a set of data tags representing targeted data elements. In some examples, the request may also specify a minimum level of assurance required for the requested data elements. Specifying a minimum assurance level rather than specifying a particular issuer allows data to be referenced from multiple issuers with the same level of assurance. In other embodiments, the request could specify a level of assurance.

[0024] As illustrated in Fig. 1, a user of mobile device 128 may request electronic credentials for his insurance coverage. A health insurance provider may request data from a government issued identification card 103, such as a driver's license. The health insurance provider issues the individual a proof of insurance card 107. The proof of insurance card 107 may reference data from the individual's driver's license 103, such as the individual's date of birth and photo. The proof of insurance card 107 may also include data issued by the health insurance provider such as coverage information 105.

[0025] In some embodiments, the aggregated credential system may verify data received from different issuers. Data may be given different assurance levels based on the data source. For example, data received from a governmental entity, such as the Department of Motor Vehicles (DMV) may be given the highest assurance level. While data received from an educational entity, such as a university, may be given a medium assurance level.

[0026] Data from issuers may be packaged to group data elements, for example, the aggregated credentials system may include a package for driver's license data, another package for boarding pass data, etc. Additionally, the data may include data tags. Therefore, the same data with the same or different data tags may be received from different issuers with different levels of assurance. For example, each issuer may include data with the data tag "name," or "date of birth," depending on the issuer the data associated with the data tag may have different assurance levels. In some embodiments, the assurance level is considered as a parameter in selecting which data/issuer to reference. In some embodiments, the new data issuer may restrict which

other issuers may be referenced. The various issuers may be identified and classified according to a level of confidence, for example, data from governmental entities may be given the highest assurance. In determining assurance levels, the verification process of the issuer may also be considered, in addition to the issuer identity. In some embodiments, issuers may sign the data they send and/or issue.

[0027] When preparing a data package that contains data referencing data from one or more other packages, the new/referencing package issuer may check the referenced data. In some embodiments, the new/referencing package issuer may verify the referenced data. For example, the referencing package issuer may check the original issuer and other parameters (e.g., data validity/expiration).

[0028] In some embodiments, an aggregated credential database may receive the request for a new credential and determine what information can be referenced from other issuers from data stored in the database, then the aggregated credential database may request the missing information from the specific issuer.

[0029] Fig. 2 illustrates an example data structure 220 to store data from various issuers. Aggregated credential database 220 may comprise any data structure suitable for storing data and issuing data in accordance with at least some embodiments of the present disclosure.

[0030] Aggregated credential database 220 may receive data packages 211-214 from various issuers. Examples of issuers include: governmental entities, educational entities, financial institutions, utility providers (e.g., cable, electric, cell phone providers, etc.). In this example, data package 211 comprises driver's license data received from the DMV, healthcare package 212 comprises coverage information received from a healthcare insurer, student ID package 213 is received from an educational entity, and boarding pass package 214 is received from an airline.

[0031] The driver's license package 211 includes data elements/data tags: name, driver's license number, date of birth, address, height, weight, photo, issue date, and expiration date. The healthcare package 212 includes data elements/data tags: name, date of birth, address, plan number, group number, member number, issue date, and expiration date. The student ID package 213 includes data elements/data tags: name, date of birth, address, student ID number, issue date, and expiration date. The boarding pass package 214 includes data elements/data tags:

name, flight number, departure date, departure time, departure city, arrival airport/city, arrival time, and seat number.

[0032] As shown, all packages include the data element/data tag name. Data packages 211-213 include the data element/tag date of birth (DOB). Therefore, if a new issuer wants to reference the name and/or DOB data element/tag, there are multiple issuers that may be referenced. The referenced data may be chosen based on factors such as required assurance level, associated cost, verification status, expiration date, etc. Although, not shown, each data package may include a unique identification. Data packages 111-114 are for illustration only, in some implementations, packages 111-114 may include additional data elements/data tags, fewer data elements/data tags, different data elements/data tags, or differently arranged data elements/data tags than are shown in Fig. 2.

[0033] In some embodiments, a data package may include and/or reference a given data tag multiple times. For example, the identical tag could come from different people associated with the same package, such as all individuals covered under the same car insurance policy. Therefore, the data tag for the policy number may be identical across all individuals covered under the same policy. In another example, the same data could be referenced multiple times in the same package. In other words, the insurance package data may indicate a primary member of the insurance package and also covered member(s), the individual listed as the primary member may also be listed under covered member(s).

[0034] In some embodiments, aggregating credentials could result in a data tag being issued and used only once for a given individual. All issued tags for a given individual could be stored in a single table and the table would contain a unique instance of each data tag. Other tables could be used to manage data packages to indicate where the data tag is referenced. In other embodiments, a table could store a data tag along with a reference to the issuer of the data, allowing multiple versions and update history to be tracked. In yet other embodiments, where level of assurance is considered, the same data tag could be used multiple times by various issuers.

[0035] With reference now to Fig. 3, details of a communication system 300 used for the generation, distribution, and use of aggregated credentials will be described in accordance with at least some embodiments of the present disclosure. The communication system 300 is shown to include the mobile communication device 328 and beacon 332. The communication system 300

is also shown to include a communication network 304 that provides machine-to-machine communication capabilities. In some embodiments, the communication system 300 further includes issuers 208 and 312 and new data issuer 316. In some embodiments, the beacon 332 may be used to provide location-based services. The mobile communication device 328 may be configured to communicate directly with the beacon 332 via a direct communication link. Alternatively, or additionally, the mobile communication device 328 may be configured to exchange information with the beacon 332 via the communication network 304.

[0036] The mobile communication device 328, in some embodiments, may correspond to any type of computing device that is portable in nature. It should be appreciated that the purchasing device 208 does not necessarily have to correspond to a portable device, but rather may correspond to a Personal Computer (PC), desktop computer, or the like. While the mobile communication device 328 is depicted as a particular type of device, such as a cellular phone, smart phone, or the like, it should be appreciated that the mobile communication device 328 may correspond to any type of device that is capable of via the communication network 304. As some non-limiting examples, the mobile communication device 328 may be provided as a cellular phone, smart phone, tablet, PDA, wearable device, portable credential card, key fob, or the like.

[0037] The beacon 332 may correspond to any type of device that is capable of communicating with the mobile communication device 328, transferring a beacon signal, and/or reading information stored thereon and/or presented thereby. In some embodiments, the beacon 332 may correspond to a device that has a fixed location, such as a point of sale (POS) device. In some embodiments, the beacon 332 may correspond to another mobile communication device that functions to read data from the mobile communication device 328 via a direct communication. The beacon 332 may be equipped with a user interface that enables data to be viewed. The user interface of the beacon 332 may include a graphical user interface, an LED, a collection of LEDs, or the like.

[0038] The credential database 320 may correspond to one or multiple servers that are owned and/or operated by an aggregated credential issuing entity. In some embodiments, credential issuing entities access the credential database 320 as part of a service. Issuers 308 and 312 and new data issuer 316 may correspond to an entity that is responsible for issuing specific credential documents. In some embodiments, the credential database 320 may be provided with functionality that enables a new data issuer 316 to request and receive data from other issuers

308 and/or 312. The new data issuer 316 may reference the received data in a new data package that includes data from the new data issuer 316.

[0039] The communication network 304 may correspond to any known type of network that facilitates machine-to-machine communications as noted above. The network 304 may use the same communication protocols or different protocols without departing from the scope of the present disclosure. The communication network 304 may correspond to one or many different networks and may also comprise any type of known communication medium or collection of communication media and may use any type of protocols to transport messages between endpoints. The network 304 may include wired and/or wireless communication technologies. The Internet is an example of a communication network that constitutes an Internet Protocol (IP) network consisting of many computers, computing networks, and other communication devices located all over the world, which are connected through many telephone systems and other means. Other examples of a communication network include, without limitation, a standard Plain Old Telephone System (POTS), an Integrated Services Digital Network (ISDN), the Public Switched Telephone Network (PSTN), a Local Area Network (LAN), a Wide Area Network (WAN), a Session Initiation Protocol (SIP) network, a Voice over Internet Protocol (VoIP) network, a cellular network, and any other type of packet-switched or circuit-switched network known in the art. In addition, it can be appreciated that the communication network 304 need not be limited to any one network type, and instead may be comprised of several different networks and/or network types. Moreover, the communication network 304 may include a number of different communication media such as coaxial cable, copper cable/wire, fiber-optic cable, antennas for transmitting/receiving wireless messages, and combinations thereof.

[0040] With reference now to Fig. 4, various examples of an electronic credential generation and issuance process will be described in accordance with at least some embodiments of the present disclosure. While the various processes will be described in accordance with illustrative steps performed in a particular order, it should be appreciated that embodiments of the present disclosure are not so limited and that any of the process steps depicted and described herein can be performed in any order or in parallel with one another.

[0041] A process is shown where a user requests an electronic credential via the mobile communication device 128 carried by the user. Thus, the process depicted in Fig. 4 represents a process where a user may correspond to the individual associated with the requested credentials.

[0042] The process begins with issuers 308 and 312 transferring data to credential database 320 (step S401). For example, issuer 308 may comprise a financial institution and issuer 312 may comprise a governmental entity. In some embodiments, the data may be transferred over a secured communication session established over the communication network 304 (e.g., a secured session using a hypertext transport protocol). The user of the mobile communication device 128, 308 may indicate request an electronic credential using the mobile communication device 128, 328 (step S402). For example, a person may request his insurance card in electronic format from his health insurance provider. The new data issuer 316, in this example the health insurance provider, requests data from the credential database 320 (step S403), which may store data from issuers 308 and 312. In some embodiments, new data issuer 316 may pay a fee in order to access the requested data. Once an appropriate payment has been received, credential database 320 transferred the requested data to new data issuer 316 (step S404).

[0043] New data issuer 316 generates the data to be issued (step S405). In some embodiments, new data issuer 316 issues a standalone package that does not contain any referenced data. In other embodiments, the issued data includes at least one reference to data from another package/issuer. The issuance process to the holder of mobile device 128, 328 of a standalone package is different from issuing a package that contains referenced data because for a package that contains referenced data, the referenced data must be available at the time of verification. In some embodiments, packages containing referenced data may only be issued to apps where the reference data package is available. The new data issuer 316 issues the data package to mobile device 128, 328 (step S406).

[0044] In some embodiments, data packages may need to be updated. If the data package references data from other issuers, the referenced data may be updated. The new data issuer that referenced the data may desire to be notified of the update and verify the updated data before referencing the data again. For example, the issuer may be notified to fetch updated date. In addition, data packages may include information about the version of referenced data. In some embodiments, the new data issuer may not want to be notified as long as the same data tags exist and accept the updated data automatically. Additionally, the referencing data package should be alerted when the referenced data element/tag is remove/no longer present. In some embodiments, delivery of a data package with referenced data could trigger an update process to deliver the referenced package to the same application.

[0045] With reference now to Fig. 5, additional details of a mobile communication device 128, 328 will be described in accordance with at least some embodiments of the present disclosure. The mobile communication device 128, 328 is shown to include one or more components, such as, a memory 504, a processor 508, an antenna 512, a user interface 520, a communications module 528, and a user interface 520. In some embodiments, the mobile communication device 128, 328 may further include a power source 524, such as a battery and one or more conversion circuits suitable to convert external power into useable power for components of the mobile communication device 128, 328.

[0046] The memory 504 of the mobile communication device 128, 328 may be used in connection with the execution of application programming or instructions by the processor 508, and for the temporary or long-term storage of program instructions and/or data. The memory 504 may contain executable instructions that are used by the processor 508 to run other components of the mobile communication device 128, 328. In one embodiment, the memory 504 may be configured to store some or all information for an issued credential.

[0047] Illustrative but non-limiting examples of a browser application (e.g., Safari, Firefox, Chrome, Microsoft Edge, Opera, Vivaldi, or the like). Instructions that enable the mobile communication device 128, 328 to communicate may include Bluetooth instructions, NFC instructions, WIFI instructions, etc. In some embodiments, the memory 504 may comprise volatile or non-volatile memory. Non-limiting examples of memory 504 that may be utilized in the mobile communication device 128, 328 include RAM, ROM, buffer memory, flash memory, solid-state memory, or variants thereof.

[0048] The processor 508 may correspond to one or many microprocessors that are contained within the housing of the mobile communication device 128, 328 with the memory 504. In some embodiments, the processor 508 incorporates the functions of the user device's Central Processing Unit (CPU) on a single Integrated Circuit (IC) or a few IC chips. The processor 508 may be a multipurpose, programmable device that accepts digital data as input, processes the digital data according to instructions stored in its internal memory, and provides results as output. The processor 508 implements sequential digital logic as it has internal memory. As with most known microprocessors, the processor 508 may operate on numbers and symbols represented in the binary numeral system.

[0049] The antenna 512 may correspond to one or multiple antennas and, in some embodiments, may be configured to enable wireless communications between the mobile communication device 128, 328 and the aggregated credential database. As can be appreciated, the antenna(s) 512 may be arranged to operate using one or more wireless communication protocols and operating frequencies including, but not limited to, Bluetooth®, BLE, NFC, ZigBee, GSM, CDMA, Wi-Fi, RF, and the like. By way of example, the antenna(s) 512 may be RF antenna(s), and as such, may transmit/receive RF signals through free-space to be received/transferred by a beacon device having an RF transceiver. One or more of the antennas 512 may be driven or operated by a dedicated antenna driver 516.

[0050] As mentioned above, the mobile communication device 128, 328 may include a power source 524. The power source 524 may be configured to provide power to the parts of the mobile communication device 128, 328 for operational purposes. The power source 524 may store power in a capacitor or similar type of charge-storage device. In one embodiment, electronics in the power source 524 may store energy in the capacitor and turn off when an RF field is present. This arrangement can ensure that energy is presented to the mobile communication device 128, 328 minimizing any effect on read distance. For example, the power source 524 may include a battery or other power source to supply power to parts of the mobile communication device 128, 328. The power source 524 may include a built-in power supply (e.g., battery) and/or a power converter that facilitates the conversion of externally-supplied AC power into DC power that is used to power the various components of the mobile device 128, 328. In some embodiments, the power source 524 may also include some implementation of surge protection circuitry to protect the components of the mobile device 128, 328 from power surges.

[0051] The mobile device 128, 328 may include a communications module 528 that is configured to communicate with one or more different systems or devices either remote or local to the mobile communication device 128, 328. Thus, the communications module 528 can send or receive messages to or from servers, beacons, other mobile communication devices etc.

[0052] The input device(s) 520 may include at least one device sensor. Illustrative but non-limiting sensors include one or more biometric sensors (e.g., fingerprint, heart rate, body temperature and/or heat signature, blood pressure, etc.), capacitive sensors, light sensors, image sensors, temperature sensors, pressure sensors, contact sensors, magnetic sensors, radio indoor

positioning sensors, location services sensors and/or devices, combinations thereof, and the like. As can be appreciated, one or more of these sensors may alternatively or additionally be incorporated into a wearable device. Yet another type of suitable sensor that can be included in the input device(s) 520 is a location sensor. A location sensor may be configured to determine a geographical location and/or position of the mobile communication device 128, 328 and/or a position of the mobile communication device 128, 328 relative to a known location. In one embodiment, this location may be based on Global Positioning System (GPS) data provided by a GPS module of the mobile communication device 128, 328. In some embodiments, the location of the mobile device 128, 328 may be provided based on cell tower data, Wi-Fi information, iBeacon information, and/or some other location information provided by a location module and/or a communications module 528 of the mobile communication device 128, 328.

[0053] The user interface 520 may or may not include one or more input devices and/or display devices. Examples of suitable user input devices that may be included in the user interface 520 include, without limitation, buttons, keyboards, mouse, touch-sensitive surfaces, pen, camera, microphone, etc. Examples of suitable user output devices that may be included in the user interface 520 include, without limitation, display screens, touchscreens, lights, speakers, etc. It should be appreciated that the user interface 520 may also include a combined user input and user output device, such as a touch-sensitive display or the like.

[0054] A network interface may also be provided that comprises hardware to facilitate communications with other communication devices over the communication network 304. The network interface may include an Ethernet port, a Wi-Fi card, a Network Interface Card (NIC), a cellular interface (e.g., antenna, filters, and associated circuitry), or the like. The network interface, in some embodiments, may correspond to a combination of the communications module 528 and an antenna 512 and may be configured to facilitate a connection between the mobile communication device 128, 328 and the communication network 304 and may further be configured to encode and decode communications (e.g., packets) according to a protocol utilized by the communication network 304.

[0055] If NFC is being used for the communication channel, then the beacon device and mobile device 128, 328 may have their interfaces/antennas inductively coupled to one another at which point the beacon device and/or mobile device 128, 328 will authenticate or mutually authenticate with one another. Following authentication, the beacon device may request the

issued credential from the mobile communication device 128, 328, or the mobile communication device 128, 328 may offer the issued credentials to the beacon device.

[0056] With reference now to Fig. 6, additional details of an aggregated credential issuance server 220 will be described in accordance with at least some embodiments of the present disclosure. As mentioned above, some components of the server 220 may be distributed among a plurality of servers or similar computational devices without departing from the scope of the present disclosure.

[0057] The aggregated credential server 620 is shown to include computer memory 604, a processor 608, one or more network interfaces 612, and a power supply 616. These various resources of the server 620 may enable functionality of the server 620 as will be described herein.

[0058] In some embodiments, the memory 604 may correspond to a computer-readable storage medium that is configured to store a number of processor-executable instruction sets. In addition to storing traditional computer programs or discrete instruction sets that receive an input and variables and provide an output based on the input and variables, the memory 604 may also be used to store one or more Artificial Intelligence (AI) programs that are configured to receive inputs and process the inputs through one or more neural networks. The AI program(s) stored in memory 604 may also be configured to be trained or otherwise updated in an autonomous or semi-autonomous manner.

[0059] The memory 604, in some embodiments, may include any type of computer memory device or collection of computer memory devices. Non-limiting examples of memory 604 include Random Access Memory (RAM), Read Only Memory (ROM), flash memory, Electronically-Erasable Programmable ROM (EEPROM), Dynamic RAM (DRAM), etc. The memory 604 may be configured to store the instruction sets depicted in addition to temporarily storing data for the processor 608 to execute various types of routines or functions. Illustrative instruction sets that may be stored in memory 604 include, without limitation, an electronic credential generation instruction 622. Functions of the server 620 enabled by these various instruction sets will be described in further detail herein. It should be appreciated that the instruction sets depicted in Fig. 6 may be combined (partially or completely) with other instruction sets or may be further separated into additional and different instruction sets, depending upon configuration preferences for the server 620. Said another way, the particular

instruction sets depicted in Fig. 6 should not be construed as limiting embodiments described herein.

[0060] Each of the instructions sets may correspond to a single discrete computer program or multiple computer programs that exchange inputs and outputs with one another. The aggregated credential instruction set 622 may enable the processor 608, when executed, to generate one or more verified credential(s) 628. The aggregated credential instruction set 622, when executed by the processor 608, may enable the server 620 to issue credentials to a mobile communication device in the communication network 304.

[0061] The processor 608 may correspond to one or many computer processing devices. For instance, the processor 608 may be provided as silicon, as a Field Programmable Gate Array (FPGA), an Application-Specific Integrated Circuit (ASIC), any other type of Integrated Circuit (IC) chip, a collection of IC chips, or the like. As a more specific example, the processor 608 may be provided as a microprocessor, Central Processing Unit (CPU), or plurality of microprocessors or CPUs that are configured to execute the instructions sets stored in memory 604. Upon executing the instruction sets stored in memory 604, the processor 608 enables various types of identity creation, management, and sharing functions.

[0062] The network interface(s) 612 may provide the server 620 with the ability to send and receive communication packets over the communication network 304. The network interface 612 may be provided as a network interface card (NIC), a network port, drivers for the same, and the like. Communications between the components of the server 620 and other devices connected to the communication network 304 may all flow through the network interface 612.

[0063] The power supply 616 may correspond to an integrated power supply device (e.g., a battery) and/or a power conversion device that is configured to convert external AC power into useable power for the various components of the server 620. In some embodiments, the power supply device 616 may include one or more devices that are capable of conditioning external power into DC current with a stepped voltage that can be used by the memory 604, processor 608, and network interface(s) 612.

[0064] The aggregated credential instruction set 622 may be configured to receive requests for data and receive data from various issuers and store the received data. Access control rules 624 manages admission privileges associated with each issued credential. The aggregated credential

instruction set 622 may also be configured to update one or more data packages 626 to reflect updated data

[0065] With reference initially to Fig. 7, an aggregated credential system 700 will be described in accordance with at least some embodiments of the present disclosure. Embodiments of the present disclosure have been contemplated in an attempt to ensure that an issued electronic credential has not improperly duplicated, stolen, or otherwise manipulated. In some embodiments, the aggregated credential system 700 includes a beacon system 736 to provide location-based services. The issued credentials may be delivered with beacon information that allows an associated application on the mobile device 728 to be woken up and notify the user 704 when a matching beacon is in range. For example, a seller located in location/building 740 may issue a pick-up receipt using the aggregated credential system 700, when the buyer 704 is in range, the purchasing application on the mobile device 728 may wake up and check in automatically verifying the user 704. In another example, a person 104 uses the STARBUCKS® app on mobile device 728 to purchase a drink at a specific store, the STARBUCKS® app on mobile device 728 wakes up when the person is in the vicinity of the selected store 740, but does not wake up when the person is near other non-selected stores.

[0066] In some embodiments, payment information may also be aggregated with credential information. For example, when payment is submitted, information such as age (e.g., for a senior discount) may also be verified. The aggregated credential server may restrict access to data, making data only available to certain issuers. Additionally, an issuer may restrict who can access/request certain data elements, these data elements would be marked to be released only to specific/verified entities. In some embodiments, the aggregated credential service may send issued data in separate packages. For example, fingerprint data may be delivered separately from the rest of the issued data for additional security. Verified issuers may be issued a certificate, which may be used to send/request data.

[0067] With reference now to Fig. 8, a method of installing and registering an aggregated credential application will be described in accordance with at least some embodiments of the present disclosure. The method begins when the aggregated credential application and data packages associated therewith are downloaded and installed onto the mobile communication device 128, 328, 728 (step 804).

[0068] The method continues with the user (or owner of the mobile communication device 128, 328, 728) securely requesting an electronic credential from an issuer via her mobile communication device 128, 328, 728 (step 808). In some embodiments, the request may indicate a minimum level of assurance and specific data tags. User identity may be verified before the request for the electronic credential may be transferred. For example, the user may use fingerprint verification or enter a password. In some embodiments, the verification may comprise a two-step authentication.

[0069] The method then continues with the aggregated credential application generating the issued data, which may include a standalone data package or a data package that references data from other issuers (step 812). For example, the aggregate credential system responds with one or more data element(s) for the requested data tags. In other words, various data packages may match the requested data. In some embodiments, the requestor could specifically request only one data element be returned and may additionally specify the issuer. In some embodiments, a response could include data elements with the same data tag from multiple packages. For example, in a request for last name + a boarding pass, the response may include boarding passes for each member of a family traveling together. In another example, a request for first and last name + car insurance cards could return all car insurance policies associated with an individual.

[0070] The aggregated credential application issues the electronic credential/data package to the mobile communication device 128, 328, 728 (step 816). The new data issuer may present the issued data grouped by issuer. As data may be referenced from one or more other issuers and the assurance level may be associated/linked to issuer. Grouping by issuers facilitates the verification of the digital signature of the issued data. Aggregating credentials allows data to be packaged/grouped by issuers and combining new data elements with referenced data elements.

[0100] Specific details were given in the description to provide a thorough understanding of the embodiments. However, it will be understood by one of ordinary skill in the art that the embodiments may be practiced without these specific details. For example, well-known circuits, processes, algorithms, structures, and techniques have been shown without unnecessary detail in order to avoid obscuring the embodiments.

EXAMPLES:

1. A system comprising:
 - a data aggregation server;
 - a communication interface coupled with a processor and facilitating machine-to-machine communications over a communications network; and
 - a computer-readable storage medium coupled with the processor, wherein the computer-readable storage medium comprises instructions that, when executed by the processor, enable the processor to:
 - request and receive data from one or more issuers for an individual, wherein the data is identified with a unique identifier, and wherein each data element is associated with a specified tag;
 - verify and store the data, wherein each data element is stored with a source tag indicating the source; and
 - prepare and present data package to issue in response to a data request.
2. The system of example 1, wherein one of the one or more issuers comprises a governmental entity.
3. The system of example 1, wherein one of the one or more issuers comprises an educational entity.
4. The system of example 1, wherein one of the one or more issuers comprises a healthcare provider.
5. The system of example 1, wherein one of the one or more issuers comprises a financial institution.
6. The system of example 1, wherein one of the one or more issuers comprises a utility provider.

7. The system of example 1, wherein the source comprises a driver's license.
8. The system of example 1, wherein the instructions further enable the processor to receive data entered by the individual.
9. The system of example 1, wherein the issued data indicates an assurance level for the issued data.
10. The system of example 1, wherein the unique identifier identifies an issuer and a source document of the data.
11. The system of example 8, wherein the source document comprises a driver's license.
12. The system of example 1, wherein the instructions to prepare the data package to issue comprises instructions that enable the processor to:
 - generate the data package that references the data received from the one or more issuers.
13. A system comprising:
 - a data aggregation server;
 - a communication interface coupled with a processor and facilitating machine-to-machine communications over a communications network; and
 - a computer-readable storage medium coupled with the processor, wherein the computer-readable storage medium comprises instructions that, when executed by the processor, enable the processor to:
 - receive a request from a requesting issuer for data from one or more other issuers;
 - generate a data package, wherein the data package comprises referenced data from the one or more other issuers and issued data from the requesting issuer; and
 - issue the data package, wherein each data element in the data package includes an assurance level.

14. A method of operating an aggregated credential application, the method comprising:
 - receiving a request for an electronic credential from an issuer, wherein the credential references at least one data element from one or more other issuers;
 - generating the electronic credential; and
 - issuing the electronic credential to a mobile device.
15. The method of example 14, further comprising:
 - receiving a notification that one of the at least one data element from one or more other issuers has been updated; and
 - updating the electronic credential

ABSTRACT

A system and method are described. In an example, the system an aggregated credential data structure that is configured to generate and issue electronic credentials that may aggregate credentials from various issuers.

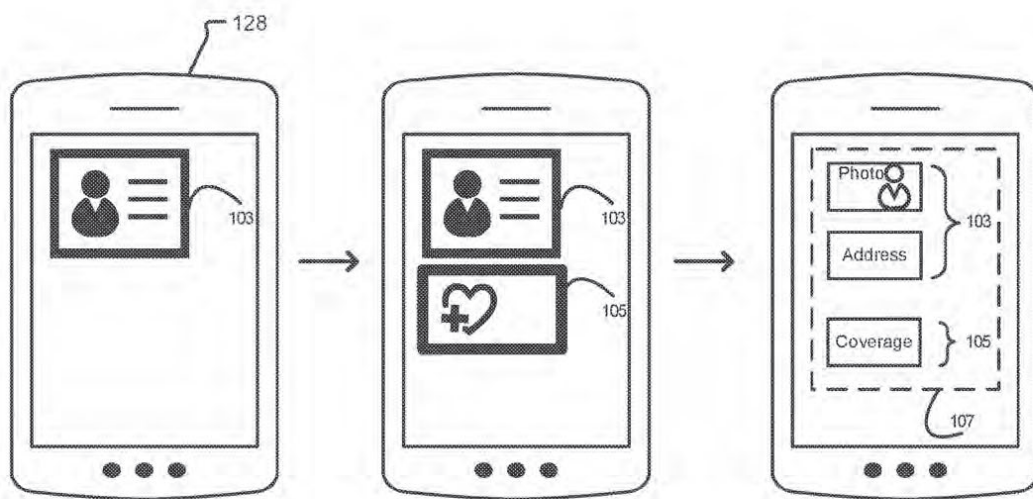


Fig. 1

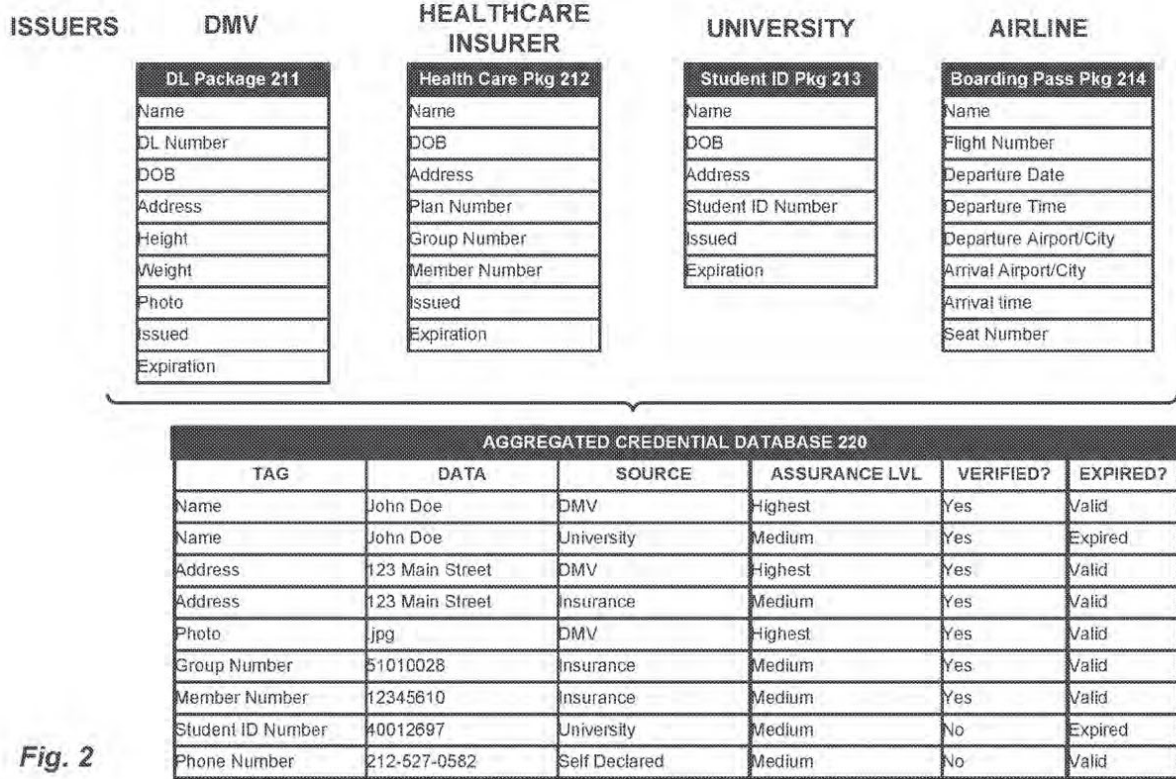


Fig. 2

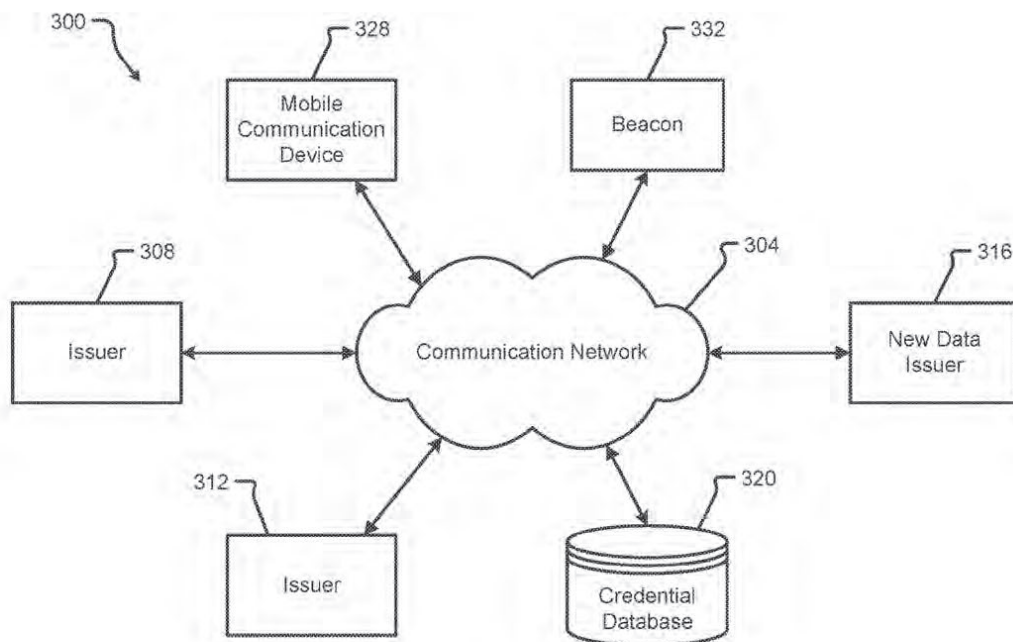


Fig. 3

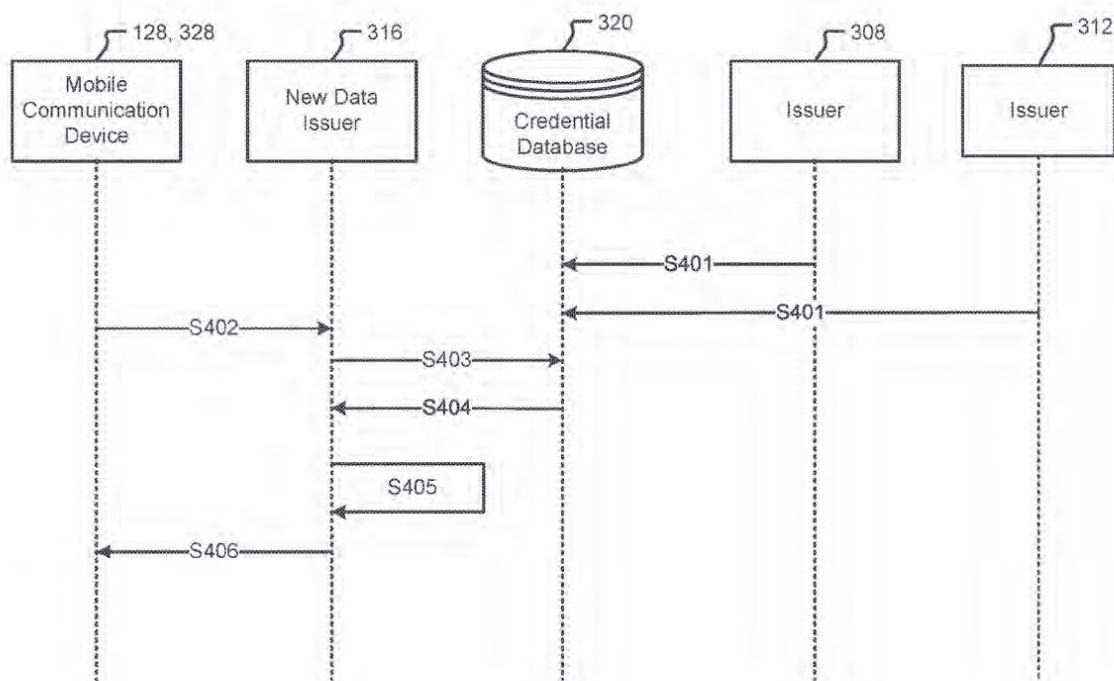


Fig. 4

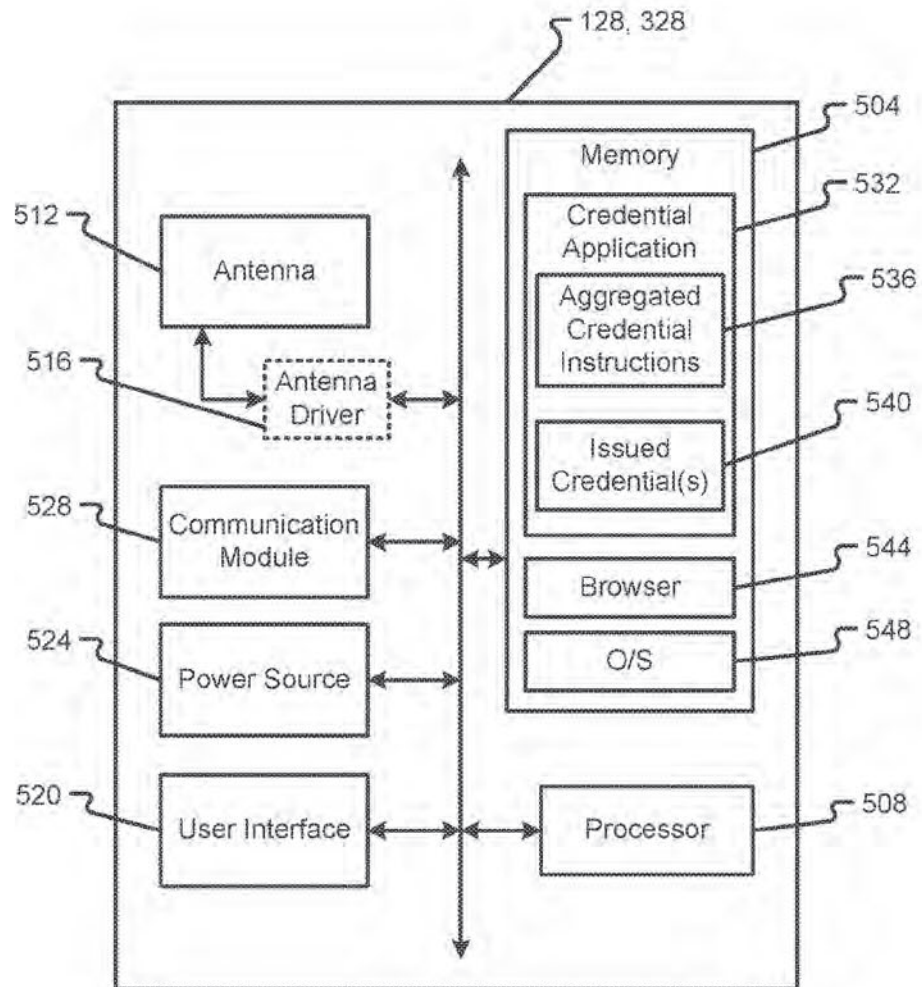


Fig. 5

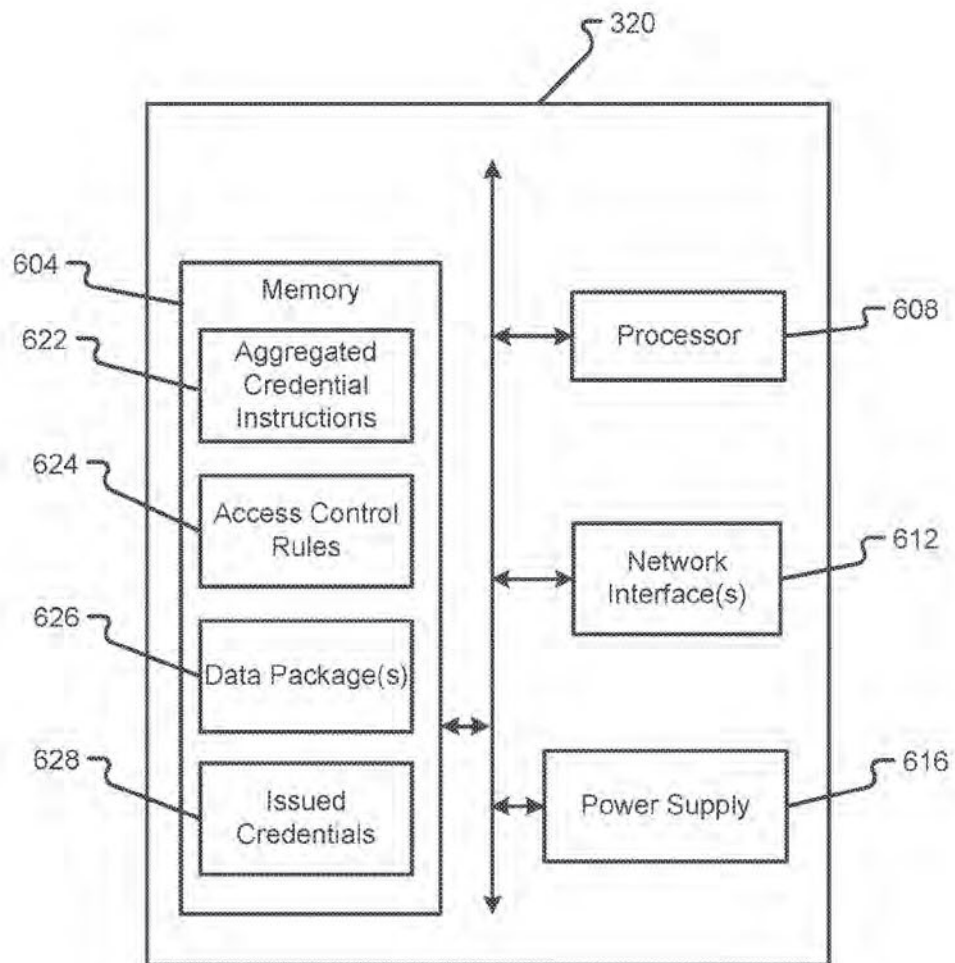


Fig. 6

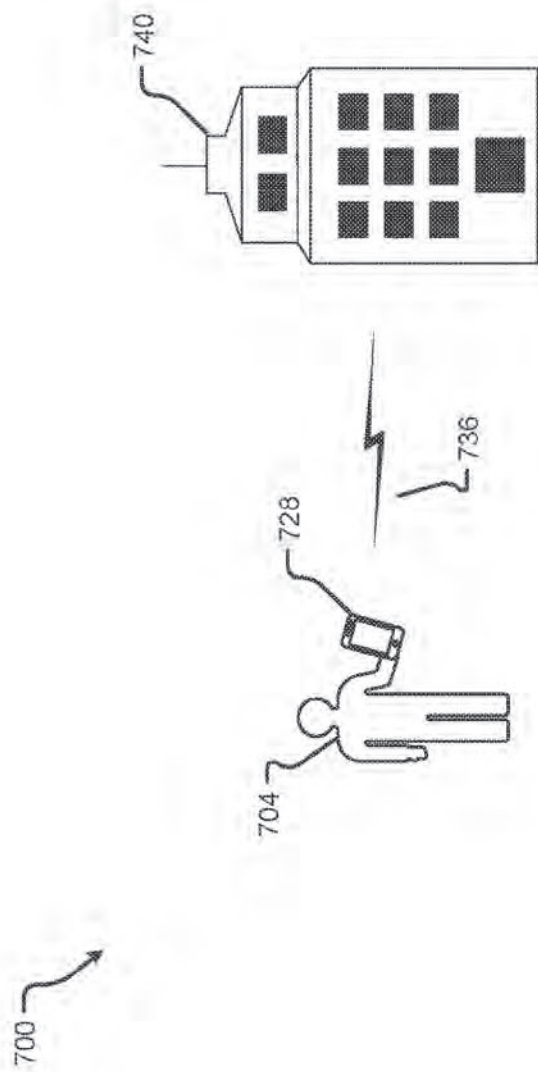


Fig. 7

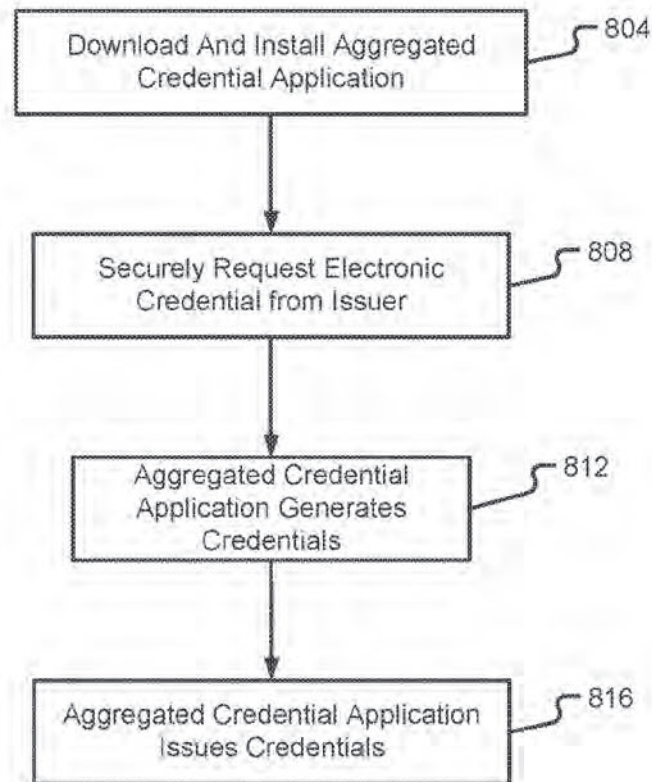


Fig. 8

Electronic Acknowledgement Receipt	
EFS ID:	35594002
Application Number:	62827626
International Application Number:	
Confirmation Number:	3071
Title of Invention:	SYSTEM AND METHOD FOR PROVIDING AGGREGATED CREDENTIALS WITH ASSURANCE LEVELS
First Named Inventor/Applicant Name:	Fabrice Jogand-Coulomb
Customer Number:	159468
Filer:	Suneel Arora/Farah O'Sullivan
Filer Authorized By:	Suneel Arora
Attorney Docket Number:	5483.145PRV
Receipt Date:	01-APR-2019
Filing Date:	
Time Stamp:	17:56:38
Application Type:	Provisional

Payment information:

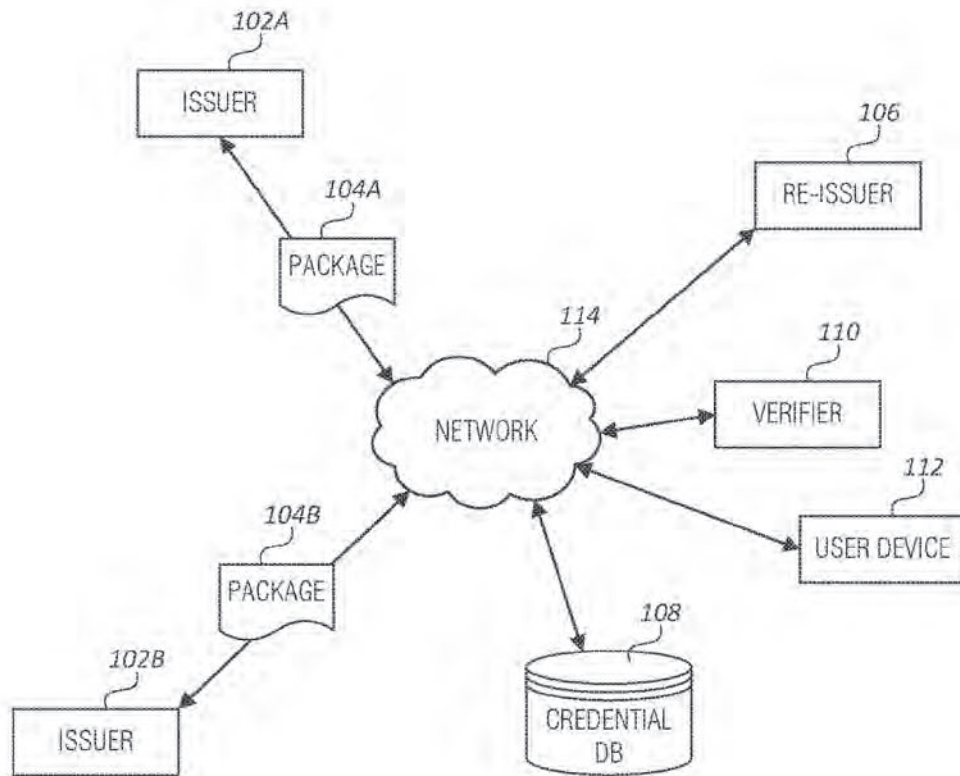
Submitted with Payment	yes
Payment Type	DA
Payment was successfully received in RAM	\$280
RAM confirmation Number	040219INTEFSW00005055190743
Deposit Account	190743
Authorized User	Farah O'Sullivan
<p>The Director of the USPTO is hereby authorized to charge indicated fees and credit any overpayment as follows:</p> <p>37 CFR 1.16 (National application filing, search, and examination fees)</p> <p>37 CFR 1.17 (Patent application and reexamination processing fees)</p>	

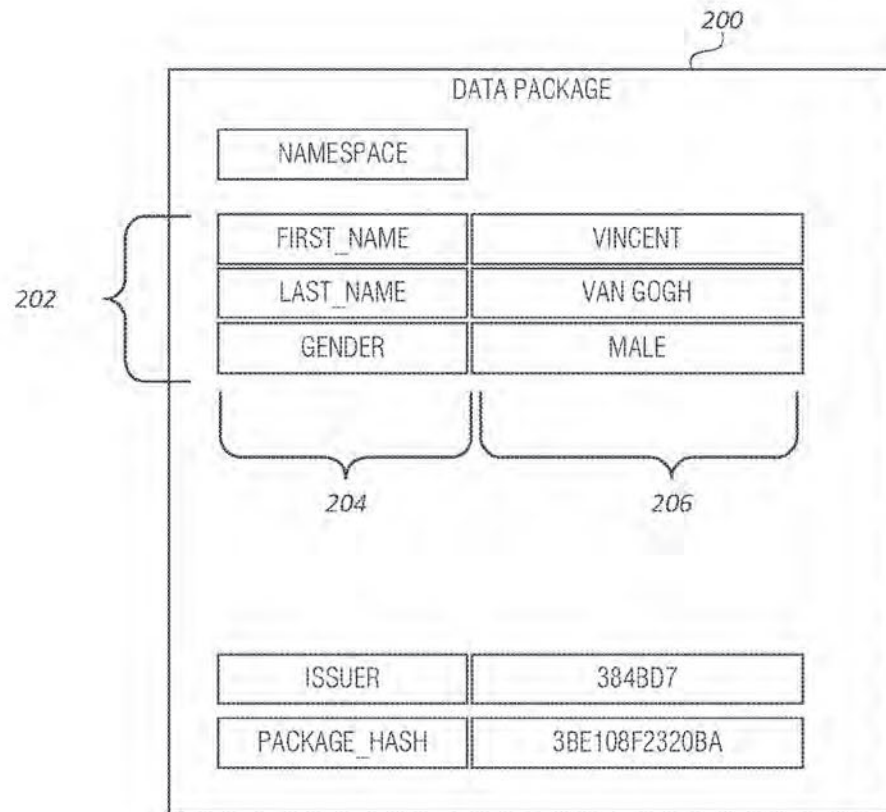
37 CFR 1.19 (Document supply fees)
 37 CFR 1.20 (Post Issuance fees)
 37 CFR 1.21 (Miscellaneous fees and charges)

File Listing:

Document Number	Document Description	File Name	File Size(Bytes)/ Message Digest	Multi Part /.zip	Pages (if appl.)
1		SIGNED_5483-145PRV_App.pdf	3132817 (110402d22f59f1359c231a0e602ab1b0b9f9278)	yes	101
Multipart Description/PDF files in .zip description					
	Document Description		Start	End	
	Transmittal of New Application		1	1	
	Provisional Cover Sheet (SB16)		2	2	
	Auth or Resc of Auth to Access Appl by DAS/PDX Office		3	3	
	Specification		4	42	
	Abstract		43	43	
	Drawings-other than black and white line drawings		44	54	
	Specification		55	62	
	Drawings-other than black and white line drawings		63	64	
	Specification		65	67	
	Drawings-other than black and white line drawings		68	72	
	Specification		73	92	
	Abstract		93	93	
	Drawings-other than black and white line drawings		94	101	
Warnings:					
Information:					

2	Fee Worksheet (SB06)	fee-info.pdf	30225	no	2
			f1a12fe3075a07240213d7e1efc3e05e15d02		
Warnings:					
Information:					
			Total Files Size (in bytes):	3163042	
<p>This Acknowledgement Receipt evidences receipt on the noted date by the USPTO of the indicated documents, characterized by the applicant, and including page counts, where applicable. It serves as evidence of receipt similar to a Post Card, as described in MPEP 503.</p> <p><u>New Applications Under 35 U.S.C. 111</u> If a new application is being filed and the application includes the necessary components for a filing date (see 37 CFR 1.53(b)-(d) and MPEP 506), a Filing Receipt (37 CFR 1.54) will be issued in due course and the date shown on this Acknowledgement Receipt will establish the filing date of the application.</p> <p><u>National Stage of an International Application under 35 U.S.C. 371</u> If a timely submission to enter the national stage of an international application is compliant with the conditions of 35 U.S.C. 371 and other applicable requirements a Form PCT/DO/EO/903 indicating acceptance of the application as a national stage submission under 35 U.S.C. 371 will be issued in addition to the Filing Receipt, in due course.</p> <p><u>New International Application Filed with the USPTO as a Receiving Office</u> If a new international application is being filed and the international application includes the necessary components for an international filing date (see PCT Article 11 and MPEP 1810), a Notification of the International Application Number and of the International Filing Date (Form PCT/RO/105) will be issued in due course, subject to prescriptions concerning national security, and the date shown on this Acknowledgement Receipt will establish the international filing date of the application.</p>					

**FIG. 1**

**FIG. 2**

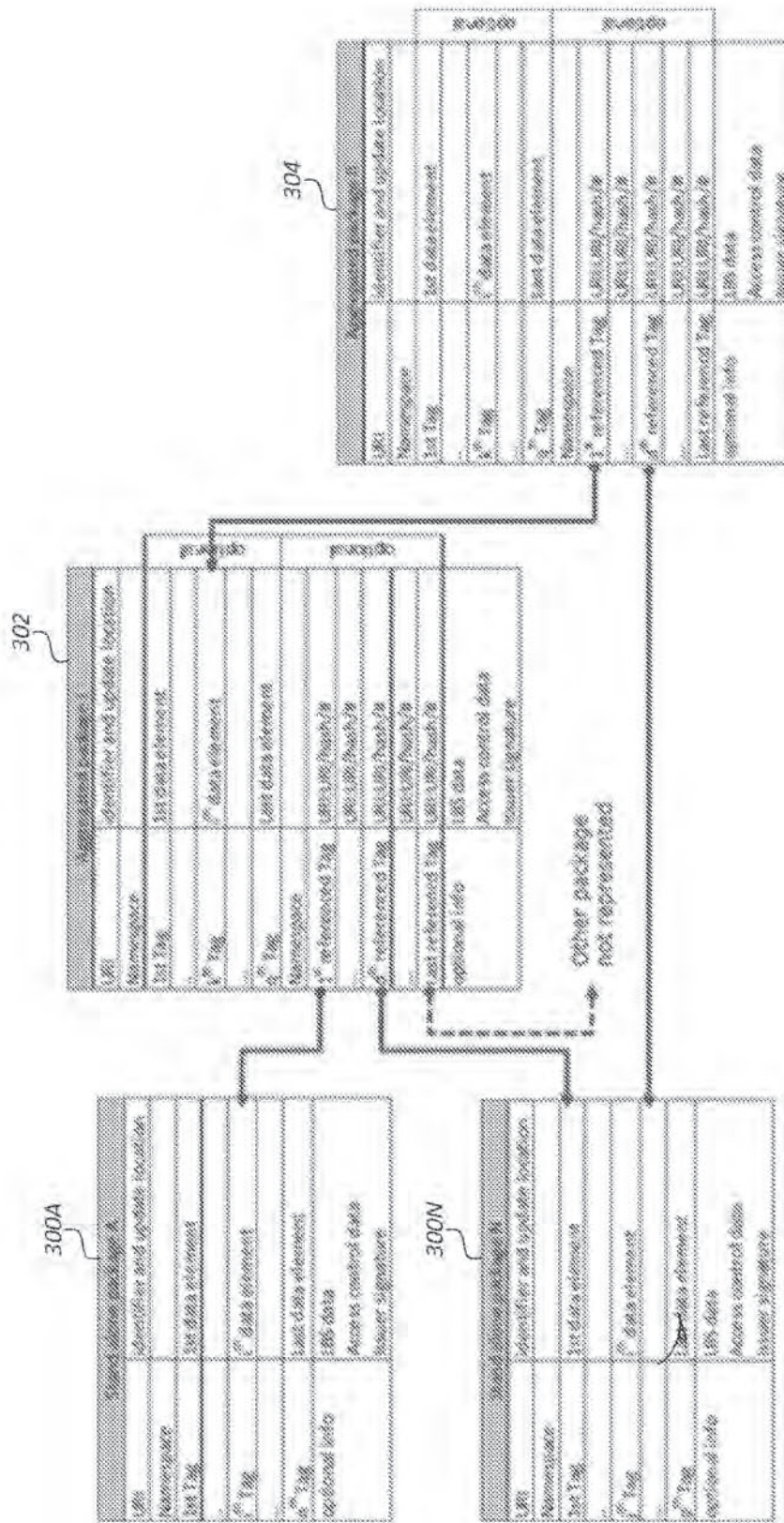


FIG. 3

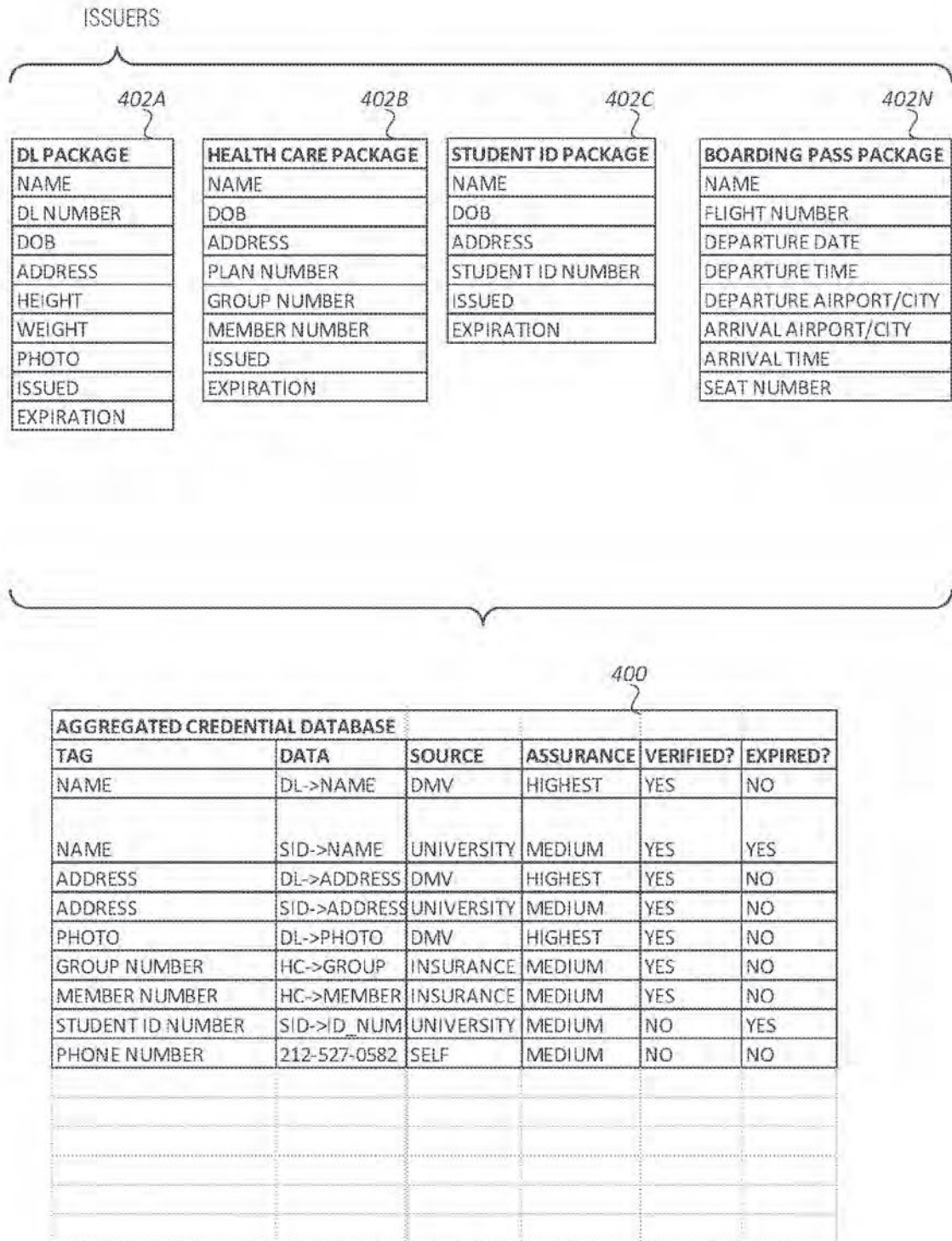


FIG. 4

5/9

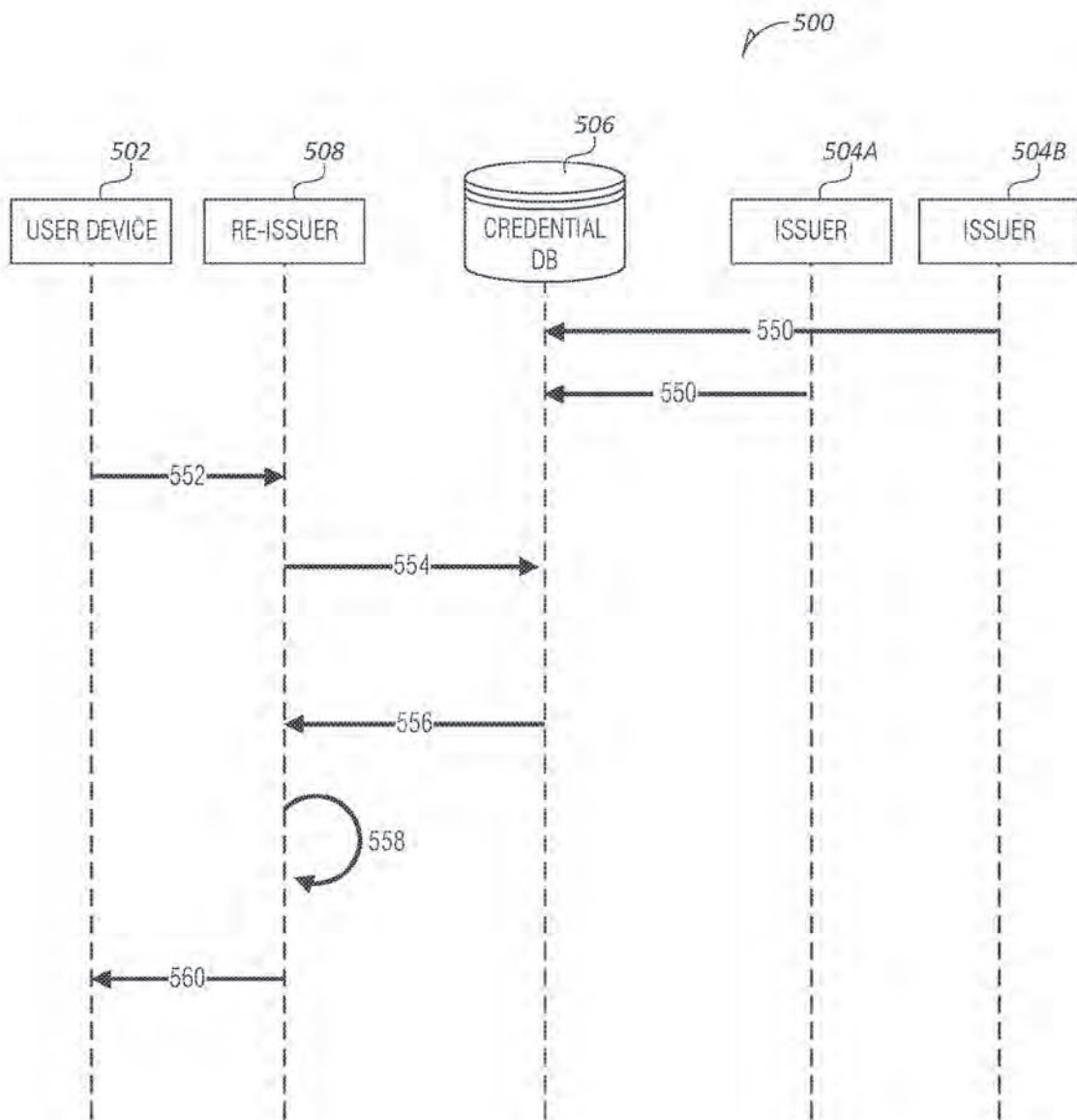


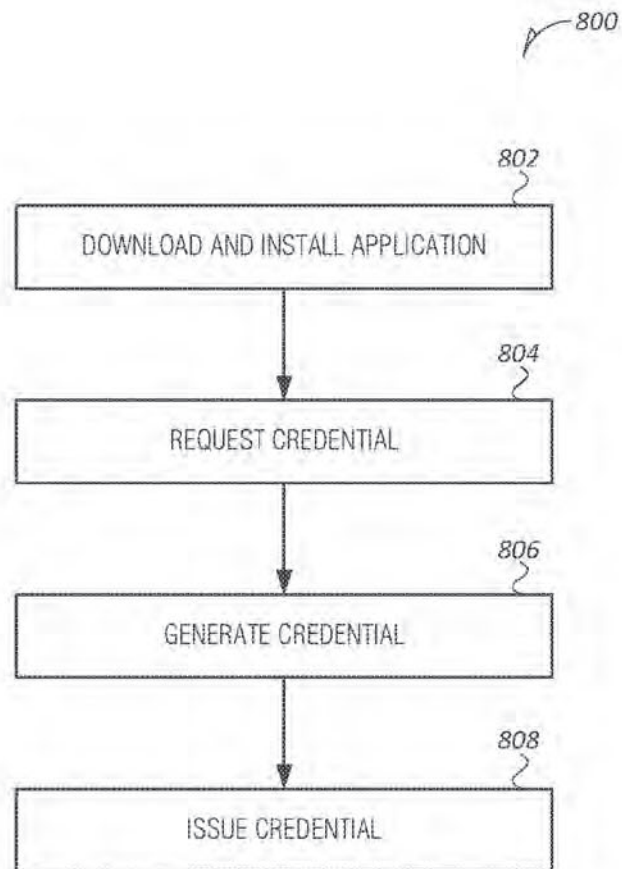
FIG. 5



FIG. 6



FIG. 7

**FIG. 8**

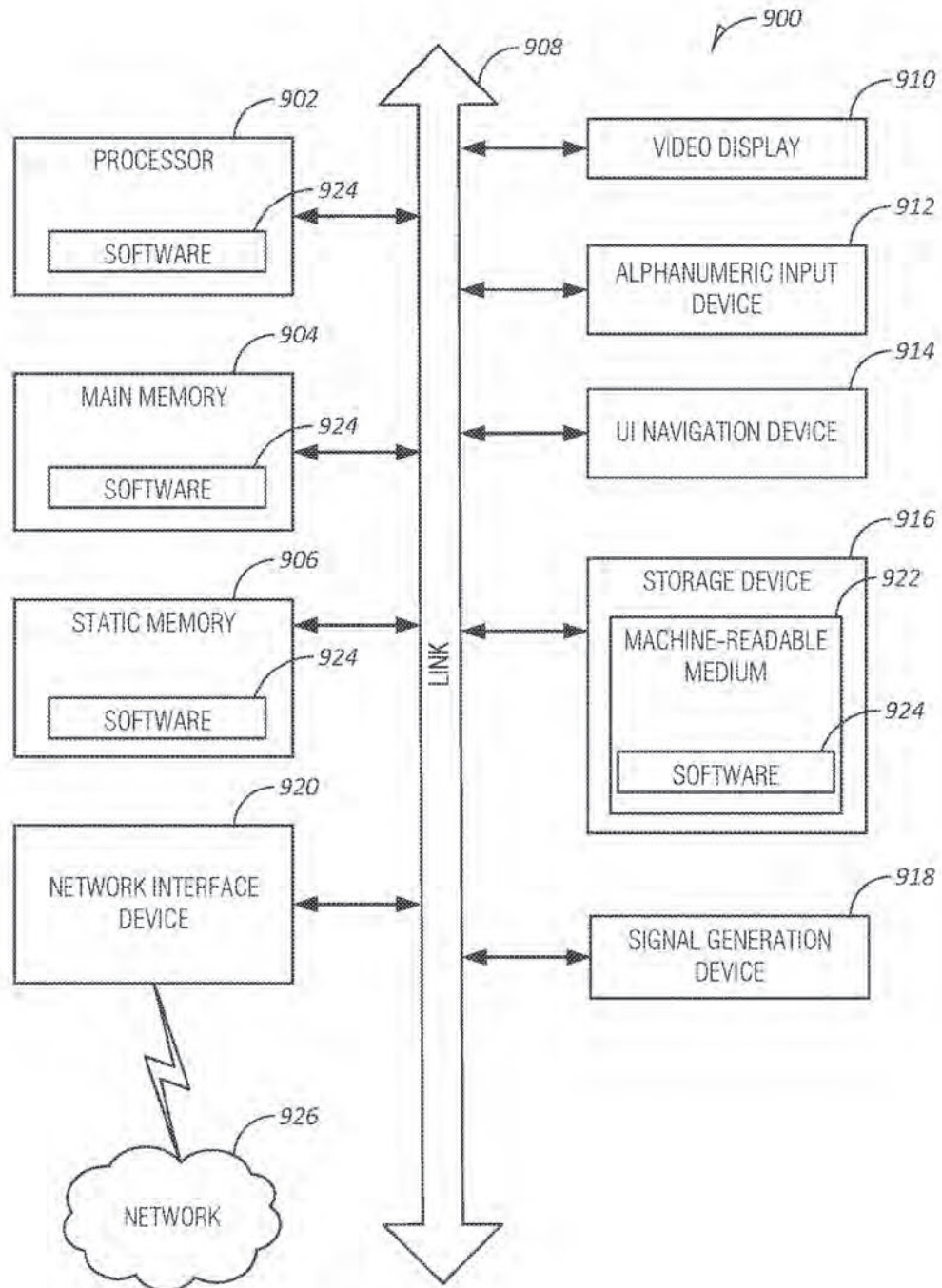


FIG. 9

The concept

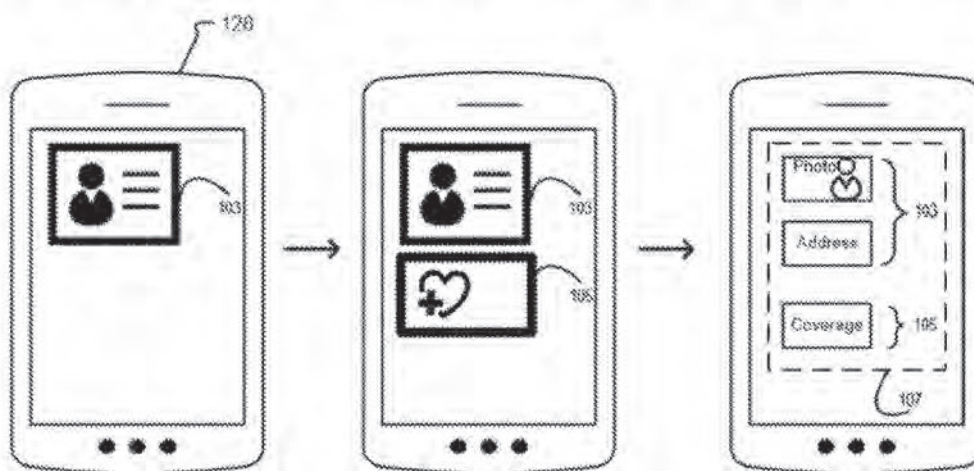
Overview

Identity document such as Identity cards have multiple information such as Name, first name, date of birth, etc. In the digital world, each of this information represents a data element and these Data element(s) are prepared by an issuer into a package that get issued to a target user.

Each data element of a given kind can be identified by a unique tag in a given name space. The name space allows to define what the given kind and its associated tag are. For example, name, first name, address, data of birth, etc. all have unique identification tags independent from the current value for a given user.

A package is typically issued to an Application that runs on a device which may be a mobile phone, a PC, a tablet, etc. The application is presented further down in more details.

A package may typically contain (new) data element(s), reference(s) to data element(s) from other packages or both. There could also be packages as well without any. Referenced data element(s) may be from packages from same issuer or another or more or a combination thereof.



A standalone package doesn't reference data element from other packages and an aggregate package contains one or more references to data elements from other packages. The end user core identity could be an example of a standalone package.

An issued package contains data element(s) each associated with the corresponding tag defined by a given name space. Using tag to identify data elements is common practice.

Packages can be aggregated to multiple packages that may be standalone or aggregated too.

A package may also contain multiple times the same tag. For example, an insurance contract may cover multiple people and the digital representation might be aggregated to the contract owner identity while referencing identity information from the other people which for example means multiple time the same tags used for name, first name, etc. In some cases, a package could also reference a tag from its own package.

A package may also carry data elements with tags from different name spaces and could also reference data with tag from one or more different name space.

Package embodiment examples

There is value in using URI for each package:

- To reference a package: it provides a mean to record where the package can/has been acquired from and what is the unique identifier of such package.
- To reference data elements from a package: the above URI can be expanded with the reference of the data element, its position in the package and for example a hash of the data element or the package the data element belongs to. Keeping track of the hash allows to detect if such package or data element has been updated.

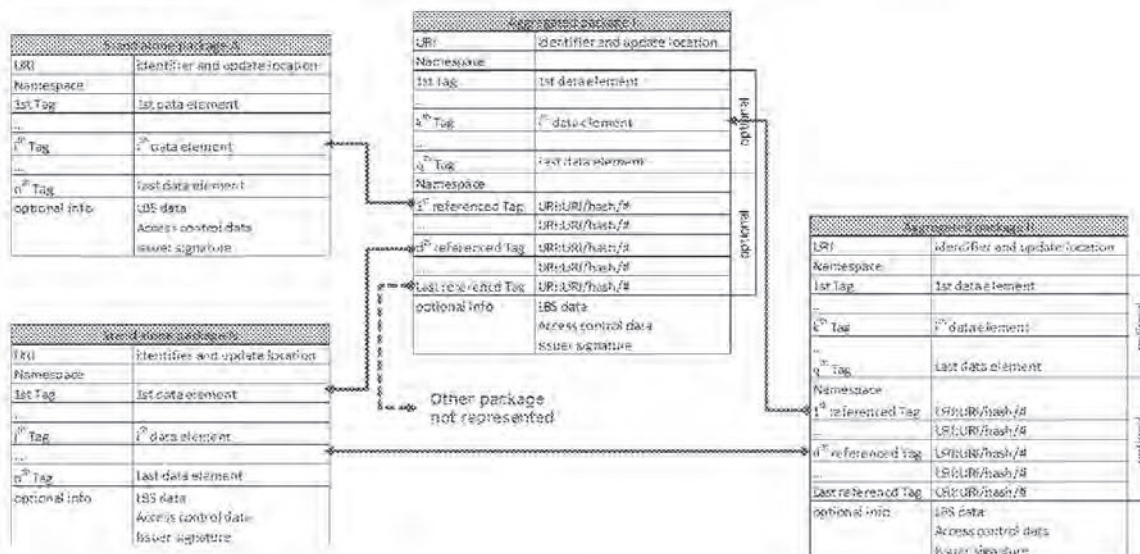
The tag of the referenced data element may not have to be referenced as there is a place for the tag of the issued – not referenced – data element that can be used for both. In addition to a tag, a package may also contain a label for each or some of the data elements which can be used for the UI. Along with a label a type of data can be specified too which allows the UI to display the information in the appropriate format e.g. date, mime type, etc.

As such a referenced data element could be handled as follows in a package:

Tag identical to the referenced data element	URI: to indicate this is not the data element value, followed by the URI per say: Download URL/Identifier of the package/package hash/#
--	---

Such approach provides a simple way for all the necessary information to handle the package and the referenced data to be available to the application. Other methods and combination exist.

Examples of packages



Typical standalone package is government citizen identity. Vehicle registration could be another example as a vehicle may be identified before it gets an owner. In that case an ownership package can be delivered with an identity that aggregates to the vehicle registration.

Aggregated package could be anything referencing an identity such as driving license, health coverage, flight ticket, etc. It could also apply to things like payment credentials which allows to check identity or age information in the same transaction as payment, access control data which allows to ensure biometrics of the right person identity can be verified during access control, etc.

In some case, the aggregation is not for a specific data element but for a kind of document whereas the idea is rather to bind a new credentials to an issued document. This could be useful for example to bind a credential used for access control to an identity. In that process the package matching the target kind of document is first identified using typical aggregated credential issuance described further down. Then credentials are issued and aggregated to the identified package.

Such binding isn't strong as enforced by the application and based on something the user own but:

- It allows to bind a credentials that could be used by other kinds of verifiers
- It is a simple enablement to ensure such aggregated credentials can be rendered obsolete when the package it is aggregated to changes or expires.

The referencing method can be similar to the one for a data element yet the URI is shortened to the package only for example as follows: URI: Download_URL/Identifier_of_the_package/package_hash

The enforcement is then handled by the application itself.

Issuance

Before an aggregated package can be issued, at least one package – standalone – is issued. Then referencing data elements can start.

The issuance of a standalone package could be quite similar to typical issuance where the package containing the data elements(s) is issued to the target user application. While aggregated credentials may all come from a single issuer, the more general case involves multiple issuers.

Standalone package

Various embodiments exist but some are of particular interest as related to the aggregated credentials: implementations where the application is from a 3rd party to the issuer. In that case, it is valuable for the issuer to confirm the kind of application and if suitable and legitimate to receive its package(s).

Many methods are available for a remote system to identify an application. These methods would not be described in here. However, along the identification, an authorized issuer can be provided with the necessary information to send notification to that particular application instance whether directly to the application or through some application backend.

Many ways are available for a package to be issued to a target application which depends in part on the user experience: for example if the request is done from within the target application then it is easy to collect the details of the application but when the trigger is received by the target user, independent from the target application (e.g. some information that triggers the acquisition is provided to the target application) then the optional identification process starts only after the acquisition trigger is received. And there are also cases where the target application is known but the instance isn't.

However, at some point the selected application from the target user starts the acquisition. Here again various cases could apply and more particularly the package can be delivered through the application backend or directly from the issuer to the application. While acquiring through the application backend is typical, the case where the application acquires directly from the issuer presents some challenges.

In particular, there is value in making sure the application connects only to legitimate address without the need to update the application for each new authorized issuer. In that case the address to acquire a package from an issuer can include some digital signature or equivalent (e.g. MAC, crypto checksum, etc.). As such the URI provided to the application and which has all the information to acquire a package also embeds a digital signature that the application can verify and connect accordingly for starting the package acquisition. Typically, the URI includes a signature from a trusted authority known to the application. For example, in a preliminary step the issuer receives general information about the application including proper information to sign the URL for the application to accept downloading.

Example of an issuer URI: HTTPS://Issuer_address.com/.../package_identifier/signature/issuer_certificate
The issuer certificate could be as short as the signed public key with a reference to the trusted authority is more than one. It is worth noting that the URI delivered by the issuer contains more information than the URL to connect to acquire a package and therefore no problem with the 2083 char limit while the URI received by the application may contain much more information that don't have to be passed to the server side such as:

- The signature and issuer certificate that is simply used for the application to see if the received information is legitimate
- The package identifier can be used in the payload of the HTTP request method used to acquire the package (e.g. the body of GET, PUT, POST, etc.).

Additional security can be put in place to ensure that only the target application can acquire the content. For example, secure delivery methods (e.g. secure channel to the application, separate channel to the end user, etc.) can be used for the URL, additional authentication/identification information can be exchanged during application discovery and used to ensure the package can only be opened by the target application instance. Additional steps can also be put in place to confirm this is the right user for example by using additional one-time code or another biometrics verification.

Finally, at the end of this process the standalone package has been acquired. And the issuer may have information to notify about updates and the application check for updates using information from the URI following a notification or alternatively regularly pulling from the issuer.

The URI of the package may be recorded by the application for later use. The URI to record or complementary information may also be delivered along with the package. And as such the URI can be signed along the package data to ensure its integrity, authenticity and relevance to the package itself.

A package may be delivered along additional information to support the user experience:

- Information dedicated to the application UI such as:
 - o a label in end user language for each data element in a package
 - o a friendly name for the package e.g. "Driving rights"

- a data type to help the UI display properly the information e.g. string, date, URL, picture and other mime type, etc.
- Information for the UI for a verification:
 - Naming a group of tags to simplify the presentation of the request where this name can be used as a purpose or as a mean to replace the group of tags instead of only showing the list of requested tag. Multiple groups of requested tags could share the same name. Having that information delivered by the issuer increases the trust level compared to receiving this in the request. And is more convenient than having it delivered as part of the application code.
- Information for related enrollment
 - For example, a package may have references or information related to enrollment for additional credentials, aggregated or not. Such information could provide a mean to enroll for additional package from within the application. The information can be delivered as a URI that contains the name of the package to enroll and a URL to the enroll site.

This is also applicable to aggregated package.

Aggregated package

In the particular case of aggregated credentials, the issuance process may vary because of referencing data elements. One preliminary step may be to check if needed data elements are already issued and are valid to be referenced. In some embodiments there is no need to access the value of such data elements and only the references are necessary – which provides value in regard to protecting PII. Validity could mean issued by an adequate issuer. Checking if the needed data elements are already issued can be either done from within the application (e.g. when filling up a form from the application) or a preliminary issuance step may involve the verification process or part of it. During that process an issuer may request for already issued data element(s) quite similarly to a request for verification. Already issued data element may be delivered accordingly or authorization to fetch from online or a combination thereof.

In some embodiment, the request could be for references to data elements instead of data elements. That could be enabled by using specific tags for the references or having a parameter in the request to mean the list requested tag is to receive the references and not the actual value.

Information about the package containing the data element is also provided, including for example the URI of the package. After confirmation that a data element is valid to be referenced, the URI of the package may also be used as part of its referencing method as indicated above.

In some embodiment, it is only necessary to receive the information to reference the data element and not the data element itself i.e. knowledge that such data exists and optionally from the adequate issuer is sufficient to be referenced. This could be the case for example when enrolling from the application itself where the application can do the necessary checks to ensure valid data elements are present.

For data element missing from the response, the issuer may choose to enroll the end user for the missing data. In that case the issuer may use its own confidence level which might be lower than referencing from proper user. Alternatively, the issuer may direct the end user to enroll with an adequate issuer for such missing data.

Another difference with a standalone package is that an aggregated package would typically be made available to the same application whose data elements would be referenced. Therefore, as the target application is typically known in advance, the issuance process of an aggregated package could be as such that the aggregated package is delivered or acquired automatically i.e. without any specific user intervention in the acquisition process. In that case the user may:

- Be Informed about the up-coming package and ask for consent before the package is prepared and not during the acquisition process
- Be Requested to confirm he/she is the right user before acquisition starts per say
- Receive the package then opt to delete if not interested

In some embodiments, the issuer of the aggregated package is provided with the necessary information to notify the source application or its gateway which triggers the acquisition of a package. The notification information can be delivered along the verification process for examples using an additional command, a special tag to request such information as part of the request for verification command or using similar application discovery as for standalone package. In the case of using a special tag, a verification process could also be used to initiate the issuance of a standalone package. The application could for example return that information only to authorized identified verifiers.

Instead of notification, the issuer of the aggregated package may also provide the information to download the aggregated package along the verification. For example, the issuer URI is provided along or after or before the request.

Finally, once the package is ready, the target application or its gateway may be notified and the aggregated package is acquired without any specific action from the end user to trigger the acquisition.

An issuer may have requested to be notified when data element(s) from one of its packaged has been referenced. In such case, the package may have additional information to ensure the issuer may be notified of such event. In that case the application uses information from the package URI to notify the issuer. For example, a PUT command is used to the specified URL and the body contains the package identifier along with information about the referencing issuer e.g. the referencing package URI.

In some embodiment, the information about aggregated packages is kept on the backend (or the cloud) and even re-issuing referenced data to a device would notify the related issuer(s). While various embodiments could exist, one option is the backend to record both the URI of the referenced package and the URI(s) of the refencing issuer(s) so that an application receiving the same package could also receive information about referencing packages and the related issuers could be notified accordingly. Notifying the aggregated issuers can be done from the application or from the backend. The backend can be the application backend or a cloud backend where the user data is saved and can be restored or transferred e.g. google drive.

Package update

The update process could be very similar to issuing aggregated credentials as the target application is known. Updating a package could be triggered by different events such as notification from an issuer, scheduled pulling from the application, user triggered pulling, etc.

The target package is acquired and the previous package with the same URI can be replaced by the new one. In some embodiments, the update process provides the hash of the current package. Then the application verifies that the acquired update references a hash that matches current state of the package to update. This is of interest in particular to prevent losing data that may have been modified through other mechanisms. For example, for transactional data element that may be updated by another system and whose information has not reconciled to the issuer backend. This is typical of transit tickets which are delivered by the issuer and where a turnstile reader may modify the ticket data and the change isn't reported in real time to the issuer or not reported at all.

With aggregated credentials, updating a package may trigger additional events. In particular when an update may impact other packages because of referenced data elements.

While in some cases, an issuer may have chosen not to be informed as long as the referenced data element still exists, typical issuers may want to be notified when:

- There is an update and the hash of the referenced data has changed or isn't available anymore
- There is an update and the hash of a package containing referenced data has changed
- The package containing the referenced data has been deleted

In some embodiments the application notifies the related issuer using information from the URI of impacted package. This is done for example by using the adequate HTTPs request method with information about the impacted package. Additional information about the referenced data elements could also additionally be provided for information.

A response to that notification could be a verification request for the issuer to analyze the data elements from the updated content.

Further events could also trigger a notification to an issuer for example when a package with a tag identical to one of the issuer packages:

- It could be of interest for an issuer to know if a more legitimate issuer has issued information that could rather be referenced in the updated packaged
- It could be of interest in case of a competitor issuer
- etc.

Advanced package

Other information could be delivered along with a package:

- Digital signature to inform about authenticity and to allow verification of integrity. It is also a mean to bind the package information together. The signature typically applies to the package but other embodiments could rely on signing part of the package such as for example limiting the signature to the data element(s) and reference(s) to data elements.
- The reference to data element could be associated with the hash of the package. This provides a mean for an issuer to find out if the source package has been modified.
- Access control information that may apply to all or specific data element(s) in a package. This provides a mean to restrict access to specific data such as finger print to certain verifiers. Such access control mechanism could rely on trusted authority where only verifier whose identification certificate is issued by one of the trusted authorities can access the restricted information.
- Location information in the form for example of GPS coordinate(s) to determine a zone or beacon identifier. This provides a mean to wake up the application when such location condition is met. Typical use cases would be Airline tickets or pick up receipt where the Application would notify the user about a verifier only when relevant i.e. when at the right gate or at the right pick location.
- Additionally, the information could be augmented with a pre-authorization for specific verifiers which would be authorize to acquire specified data without the need for user consent.
- Authorization to re-issue an aggregated package to another user or entity. For example, to transfer a pick-up receipt for another person to pick up.
- Authorization to transfer the information to another application e.g. share to another user or share to change application.
- A package could also be delivered with some expiration independent from the expiration date of the document. Such expiration can also be enforced by using tokenization.

An issued package may contain no data element or references. This could be used for example to deliver information for the application to register to wake up for specific events, locations or other conditions.

Furthermore, temporary package might be delivered empty except for the friendly name which in that particular case may indicate a pending update/delivery. The actual name of the package may be updated along the acquisition of an update.

Re-issuance/ derived issuance

Using URI to identify package and reference data element also provides additional benefits in regards to re-issuance. One aspect is to inform the issuer whose package data has been referenced and by which issuer(s). Accordingly, the issuer of the referenced package can notify the identified aggregated issuers when the same user gets the package reissued or similar package issued. It also provides a mean to notify such issuers when similar package is issued to other users.

Another aspect is for an end user to authorize another user to act on their behalf. This could be for example for an end user to allow another user to use a pick-up receipt on their behalf. In that case the package gets aggregated to another user identity. This could be done either by aggregating another identity to the existing one or by replacing the identity by the new one that the verifier should check. That

process is initiated by the end user the package has been delivered to and may go through a backend system that re-issue for the new target user.

Another aspect is for an end user to receive a package for another user. This could be for example for tickets from the same booking. In that case, it might be valuable for the user to also receive the information to aggregated to so that all information can be presented at once. The aggregated information comes with the URI to acquire the missing referenced data. In that process authorization for the referenced data may have been delivered to the issuer so that the application from another user can acquire the information. For example, the referenced data application has delivered an authorization for the target end user or its application.

In some embodiments, verification can be implemented as a re-issuance instead of the typical process of responding to a request by tag. In that case the response could look like a package issued by (or on-behalf) of the holder application and aggregating information from the various packages whose requested granted data elements belong to.

The application

While various implementations of the application exist, it is typically made of one or more software components and has the capability to communicate with issuance system(s) and verification system(s) and that:

- Can receive or acquire packages and may receive updates from one or more issuers.
- Can receive a request for verification containing a list of one or more tags each identifying a unique kind of data element and return a response
- Has either access to the data elements to return in the response or can provide an authorization to access such data elements or a combination thereof.
- Optionally feature some user interface
- Could have additional features or interfaces with the holder device such as to enable Location Based Services (LBS).
- Optionally features a notification interface

Such application can be implemented in many different ways including but not limited to simple HTML 5 web page received from the application provided and running at the holder and native application installed from typical app store.

Such application can run on a variety of devices such as mobile phone, PC, tablet, wearable device, etc. Some implementations may also split the application on multiple devices. For example, the package acquisition agent can be implemented on the mobile phone while the verifier interface is implemented on a wearable and where as the user interface can be present on both devices.

Some components of the application can also be implemented in a backend system instead of the holder device.

Such application provides interfaces to the different systems and received packages can be stored in or out of the application itself as long as it is available to respond to a request.

Once a package is received, the application can keep track of the download URL and use this for checking for update. While the application can build the URI to record based on a received download URL and package identifier, an URI may be delivered with the package itself which could serve for

- Providing a unique reference of the package
- Providing an URL to use for various commands
 - o Check for update e.g. HTTPs GET to download the package or an update
 - o Check for up to date status e.g. HTTPs POST token ID to get a status
 - o Notify an issuer referencing information that an update has been acquired e.g. HTTPs PUT to the referencing issuer with URI and hash of the referenced data that has changed.
 - o Inform an issuer when a package has been deleted e.g. HTTPs DEL which might contain information about the App where the package was provisioned.
 - o Inform an issuer about an aggregation for example HTTPs PUT with information about the package with referenced data and the URI of the aggregated package.
- Such URI may also be used to reference data element by adding the referenced tag to it and some information to ensure the matching data element is uniquely referenced such as the hash of the value of the referenced data element

Verification

Verification is about requesting data element(s) by tag(s), then proceeding with the verification per say. Additional information may be added to the request such as: specifying a document type, a confidence level, terminal identification, etc.

Typically, the verifier sends a request containing the list of tags and the response comes back with data elements matching some or all of the requested tags. Additional complementary information could be delivered too. The returned data is typically organized by “documents” as this is how the digital signature of the issuer applies. The verifier checks the validity for each document as for example: integrity check for the received data elements, authorized issuers, expected type of document, meet target level of confidence, etc.

In the case of aggregated credentials, a document per say may rely on information from different packages which may be from other issuer(s). As such while the response format could remain similar with packages returned in place of the documents, the part of the verification process where the data element(s) is(are) verified is slightly different and the UI might showcase the various issuers involved in a single document.

In some cases, the trust is implicit: if the verifier trusts the issuer of the aggregated package then it doesn't matter the trust of the referenced data and only the integrity is verified i.e. regardless of the issuer being on the trusted list. In some other cases the trust is explicit and then all issuers are verified as trusted.

There is also information that is present on many packages such as expiration date. In some embodiment, such information can be returned automatically along with data element or referenced data element from the same package. In other embodiments each have a unique tag yet such unicity could be built on a generic base for the kind of data and adding specific information related to the package or the kind of package e.g. base is exp_date and exp_date_DL for expiration date of a driving privilege package.

Advanced request

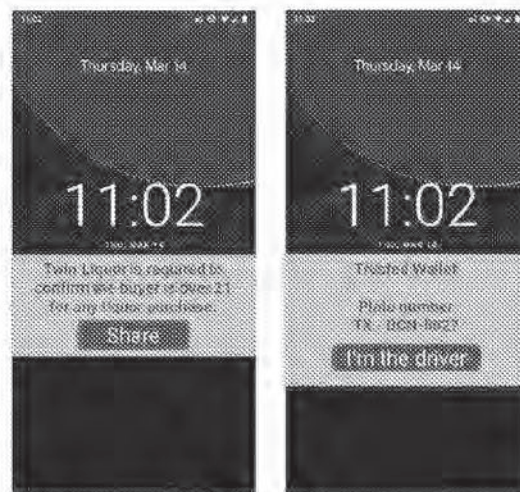
In some embodiments, in addition to the list of tag(s), the request includes a query (e.g. SQL) that aims at improving the user experience by facilitating the selection of data elements from different packages based on criteria defined in such query. However, as the list of requested tags is provided separately, the SQL request is aimed to specify the expected relationship and a response might be provided even if some data from the SQL is missing. For example, the SQL could be used to specify the expected value of a requested data element: this could be useful to request data elements for a local purpose such as pick up receipt or flight ticket. For example, the SQL could be used to specify the relationship between different elements such as car registration whose class of vehicle is matching with the corresponding tag from a driving license and where the card insurance is matching both the vehicle and the driver.

The SQL could also be used to quantify information not in the data elements such as indicating a minimum level of confidence or specify a given issuer.

From a user experience point of view, the SQL sequence can be used to drive the various steps for a user to select adequate packages for the response: Following the SQL sequence, packages that are not adequate are filtered out then the filtering process continues following the selection of first data elements or package reducing further the choices based on pre-selection. For example, the SQL query can specify that tags for vehicle category from car registration should match with the vehicle category from a driving license then that car insurance should match both car registration and identity information the driving license is aggregated to.

In some embodiment, the request could be received as different steps where one step is a broadcast which could be generic or targeted and implemented as multiple steps e.g. beacon to wake up the application then application acquires more details. Then another step where the application might notify the user and could communicate with the verifier to receive all or the remaining details of the request using the same process as for a typical verification. For example, the first step could be for the request to notify a driver of a car by broadcasting the information about the plate number. Then application without data matching that information ignore the broadcast where as other application could connect to receive the second part of the request and notify the notify. Or notify the user then connect accordingly for the continuing with the request/response process or just for sending the response.

For these implementations, a first part of the request could contain information for the application to decide if the request should be ignored or not, information about the notification information and about the call for action (e.g. for the button to click). Then the application could connect to get the second part of the request which could bring the specific tags and relationship information. Or the application could connect after the call for action. As such, depending on the process, the notification can close and the process continues on the verifier side. Or the notification opens the application where the second part of the request can be received and where the user could have more control on what's being shared for example by confirming PIN or biometrics and/or see in more details the content of the request or what is going to be returned.





for access control to the application then



to show the information

to return that is matching the request

Once the data elements have been received and the issuer identified, some or all of the information can be recorded in a blockchain while both the verifier information could remain anonymous as well as the end user. This could permit the issuer to verify the number of transactions and claim a revenue share from the cumulated transactions. In that particular case, the information that needs to be reported is the URI of the package whose data element has been verified – no need to report the verifier or information about the end user information. That being said, in some embodiment where the application checks the blockchain for verification, information about the verifier could be reported as well.

There are also special cases where multiple packages with tag matching partially or entirely the request are returned at once. A typical use case would be for flight tickets: the boarding gate sends a request for particular flight ticket data and the response comes back with data for multiple tickets.

Reporting back to the end user

Taking advantage of the URI and notification an end user can be notified every time his/her information is presented to a verifier. The URI associated with the package provides a mean for the adequate issuer to be notified and such issuer could use the notification mechanisms to inform the end user through the application.

Such process enhances the overall security and provides more confidence in the system.

Optimum implementation example

Installation of the application

In a preliminary step, the end user is informed about an adequate application(s) to use for the credentials of a given issuer. Installation of the right application can be driven from the list of adequate application which may also embed links to install the application from the adequate app store.