STREAMLINE BORDER CONTROL WITH BLOCKCHAIN TOWARDS SELF-SOVEREIGN IDENTITY

Pekka Koskela, Anni Karinsalo, Jori Paananen and Laura Salmela

VTT Technical Research Centre of Finland Ltd,
P.O. Box 1000, FI-02044 VTT, Finland

ABSTRACT

Since the mid-2000s, the digitalisation of border checks has often referred to the increased adoption of automated border control (ABC) solutions at border crossing points in all border environments from air- ports and seaports to land border crossings. Key prerequisites for the operational implementations of the so-called eGates have been the electronic machine-readable travel document together with biometric technologies that have facilitated the automation of much of the tasks performed by border guards at manual control booths for selected groups of nationalities. Now, the next wave of major changes is emerging with the development of electronic identification (eID), with certain implementations particularly designed for cross-border use cases supplementing and possibly replacing the traditional physical identity document in a long-term future. The evolution of eID strongly aligns with the increased demands for data privacy to ensure that individuals can better control how much information is shared about themselves, with whom and for what purpose. One possible technology to provide the so-called data self-sovereignty is distributed ledger technology (DLT), including blockchains. DLT is being developed for instance by the Linux foundation, dispensing several distributed ledger projects and associated solutions for digital and self-sovereign identity. One of these projects is Hyperledger Indy. In this study, we present a distributed ledger implementation based on Hyperledger Indy applied as a border check use case. Our aim is to investigate the suitability of DLT in providing data self-sovereign facility in border checks, and to discuss the benefits and disadvantages the technology might entail for this security domain.

KEYWORDS

Blockchain, Self-sovereign Identity, Border Control.

1. INTRODUCTION

The history of automated border control (ABC) at EU's external borders dates back to the late 2000s, when the first pilot systems appeared at border crossings points (BCPs) of a handful of Member States, such as Finland, France and Portugal [1]. Approximately 15 years later, the original objective of ABC remains unaltered despite the drastic effects of the COVID-19 outbreak on international travel (more on impact see e.g. [2]). Essentially, ABC systems provide a technological solution for striking an acceptable balance between passenger flow facilitation and the provision of sufficient security level against various threats in the context of growing traffic volumes that exceed border authorities' infrastructural capacity and processing capabilities [1]. Recent advances in ABC emphasize modularity and non-stop checking approaches that support better system adaptation into different operational environments and improve user experience by eliminating the mantrap from single gate infrastructures [3]. Now, a new concept, the Digital Traveller Credential (DTC), is making headway and potentially securing the achievement of a

David C. Wyld et al. (Eds): SIGV, AI & FL, SESBC, MLAEDU, DSCC, NLPTT, SCOM, SCM, CEEE - 2022
pp. 75-92, 2022. CS & IT - CSCP 2022 DOI: 10.5121/csit.2022.122207

truly frictionless and also contactless border crossing, at least in airport scenarios. Besides including identical data of the physical travel document, additional attributes could be included to the DTC, such as health certificates or an electronic visa depending on the needs and requirements of individual travels e.g. [4] [5] [6].

The DTC standardisation effort, initiated and coordinated by the New Technologies Working Group (NTWG) of the International Civil Aviation Organisation (ICAO) [7], temporally coincides with the general progress on digital identities within Europe. In September 2020, President von der Leyen underlined the need to create a secure European e-identity in her State of the Union Address [8]. In March 2021, the European Commission (EC) published the 2030 Digital Compass outlining a roadmap for the acceleration of digital transformation, including also targets and milestones for broader digital identity implementation [9]. This was followed by the EC's proposal in June 2021 for a European Digital Identity (EDI) framework that would enable citizens' access to various services by using their national digital identification implemented as a digital wallet application [10]. Simultaneously, the EC recommended Member States to cooperate in the development of a toolbox for practical and coordinated adoption of the EDI framework [11].

One key pillar of the EDI framework is user controlled digital identity [12]. This aligns with the concept of Self-Sovereign Identity (SSI), according to which individuals can control their digital identities and have true control over that digital identity, creating user autonomy. SSIs are perceived as the next generation of digital identities across open networks enabling cross-border identification while complying with existing and shortly revised regulations, such as the General Data Protection Regulation (GDPR) [13] and the electronic IDentification, Authentication and trust Services (eIDAS) [14]. A move towards digital IDs can also be examined from the lenses of 'bring your own device', as the services concentrated on a mobile device constantly expand with people familiarizing in their use and expecting their availability (see e.g. [15]). As claimed by [16], there is a shift from treating digital identity 'in a predominantly sectoral fashion whenever necessary' to 'a basic infrastructural service, sometimes even a commodity'. To advance the SSI development, various EU projects investigate the potential of the distributed ledger technology (DLT) in providing privacy-preserving, secure and trusted solutions [17]. As part of the European Blockchain Services Infrastructure (EBSI), development work is provided as a particular use case for SSI. This use case demonstrates that EBSI can implement cross-border verification of identity credentials allowing users to create and control their own identity across borders [18]. In general, DLTs, and blockchains utilizing the DLT architecture, are well-suited for implementing SSI systems. Blockchains' distributed architecture without a central authority, and the fault-tolerant consensus mechanisms support the trust needed in the SSI systems [16].

The border crossing context offers a suitable environment to advance the SSI development, as the interaction processes between travellers and authorities rely on using and authenticating user identity. Strict but clearly defined security and functional requirements and roles, and a justified need for smooth but still secured processes also support SSI development in this field. In this paper, we present our research related to the concept of SSI and its operation in the border control domain. Our contributions are:

- Presenting potential use cases of SSI in border control process
- Implementing Hyperledger Indy mobile and server applications
- Providing test results of the previous, that were validated by the border control officers
- Providing further discussion of the topic

These are presented in the following sections as follows:

Section 2 reviews current literature on the topic, while section 3 presents the concept of SSI and principles of technologies relevant to our implementation. Section 4 describes border crossing point use cases. Next sections 5 and 6 explains our implementation details and test and demo setup. Section 7 provides discussion about benefits and down sides of our solution, and finally, section 8 provides conclusions.

2. SSI and Blockchain Technology in the Border Security Domain

Although there is abundant literature on the implementation of blockchain technologies in various fields, documentation on real-life applications of blockchain-based digital identity solutions appear somewhat limited in the border security domain. Recorded examples of particularly large-scale applications are scarce, and individual studies primarily concentrate on the airport environment. For example [19] have examined the overall applicability of blockchain technologies to the aviation industry. The study identifies identity management as one of many potential blockchain use cases. However, no interlinkage is made with governmental immigration processes, as the main focus is on the perspectives of commercial actors (i.e. airlines). In their review article, [20] refer to the Known Traveller Digital Identity (KTDI) concept of the World Economic Forum which intends to serve passenger identity management. KTDI's technological background rests in cryptography and distributed ledger technology. To facilitate the concept's demonstration, a set of standards, specifications and best practises has been published [21]. Also, other milestones have been achieved since the beginning of the collaboration process in 2018 (e.g. development of a dedicated KTDI mobile application). At the moment, the KTDI pilot between various governmental and commercial partners, such as the government of the Netherlands, Accenture, KLM Royal Dutch Airlines and the Amsterdam Schiphol Airport, has been suspended due to the reopening of global travel after the COVID-19 pandemic. However, continuing cooperation in the area of digital credentials is considered as highly important. [22].

In comparison, [23] report on a more limited application of blockchain technologies for the implementation of digital passports at the Dubai International airport. The "Emirates Smart Wallet" uses traveller's personal ID, passport details, and smart gate card data and connects them with e-gate services to facilitate immigration clearance. [24] "The second phase will link all the data of Emirates and residents into the wallet so people don't need to show their documents when transacting in any government department." [25] The smartphone application generates a bar code that is scanned in a gate. [26] present a blockchain solution for travel document issuance process. The implementation is based on public blockchain of the Ethereum platform.

SSI solutions can also be implemented with non-blockchain platforms. Bissessar et.al [27] have proposed a non-blockchain based credentials that are stored on a smartphone solution. They present credential where information of the travel documents is saved in a digital form in the smartphone, and the information is secured by cryptographic operations that exploit privacy-respecting facial biometric references. The issuance of credential is done over internet to issuance server. Another non-blockchain based solution proposed by Patel [28] simulates passenger processing of airports and biometric single token identification (ID) to enhance the process. The token ID is perceived as enhancing the process while ensuring safety and security. The token ID can be considered conceptually the same thing as digital credentials described in the use case above.

As the infrastructural requirements in non-blockchain solutions are low, their implementation can be understood as being simpler. However, non-blockchain solutions similar to the above do not support the generic self-sovereign concept comprehensively. Implementing SSI may require different kinds of credentials and even implementation beyond state borders, which may not be possible for a non-blockchain solution. Additionally, non-blockchain solutions usually include neither cryptographic assurance mechanisms based on block structure, nor distributed architecture.

3. SSI AND RELATED TECHNOLOGIES

In the following, we describe the concept of SSI as well as the main ideas of the Hyperledger Indy and Verifiable Credential Data Model, which are broadly used open source technologies for constructing SSI applications, and upon which our implementation is also based.

3.1. SSI

Academic literature discusses broadly the concept of SSI and defines it in various ways. However, there are certain common properties associated with the concept originating from the work of Allen [29]: Existence, Control, Access, Transparency, Persistence, Portability, Interoperability, Consent, Minimalization and Protection. In other definitions, these properties then have been further organized, such as by Sovrin Foundation, into groups [30] of Security, Controllability and Portability or such as by Bissessar et al. [27], by adding groups of Foundational, Flexibility and Sustainability into those of Sovrin. In the latter, they define the SSI of a person to be "the union of her different partial identities in each of different decentralized domains." This addresses a crucial part of the SSI. It implies that user’s identity can be built granularly, in terms of the exposed data, within different systems and, and also vary between all these systems.

3.2. Hyperledger Indy

The Hyperledger Indy (Indy) is an open source digital ledger with blockchain(s) platform provided by the Linux Foundation. Indy provides decentralized digital identities rooted on blockchain and supports self-sovereign functionality. Indy also provides Verifiable Claims which are an interoperable format for exchanging of digital identity attributes and relationships currently in the standardisation pipeline at the W3C. The Claims use Zero Knowledge Proofs (ZKP) [31] for proving that some or all of the data in a set of Claims is true without revealing any additional information, including the identity of the Prover. The ZKP forms a framework, in which one can manage what information is needed to be shown during claim verification process. Indy also includes link secrets for avoiding to leave any breadcrumbs between the two credentials and preventing the verifier to track anything one is doing.

Indy uses Redundant Byzantine Fault Tolerance (RBFT) consensus protocol to agree the transaction order, where multiple parallel instances of BFT (Byzantine Fault Tolerance) protocol are operating. Every instance will have a primary replica and one of those acts as a leader, determining the order of transactions and communicating it to the rest of the replicas. If the leader replica performance decreases due to malfunction or malicious behaviour, a new leader will be selected.

To enhance the operation and to support many types of transactions, Indy, like also many other ledgers such as Ethereum, save states of different transactions which are maintained by the Merkle or Merkle Patricia Tree.

3.3. Verifiable Credentials Data Model

The operational architecture model of Hyperledger Indy is based on Verifiable Credentials Data Model, see Figure 1, where the actors are:

- Issuer creates a verifiable credential form, which is transmitted to a holder.
- Holder possesses and generates presentations of verifiable credentials.
- Verifier receives a verifiable credential from the holder for processing.

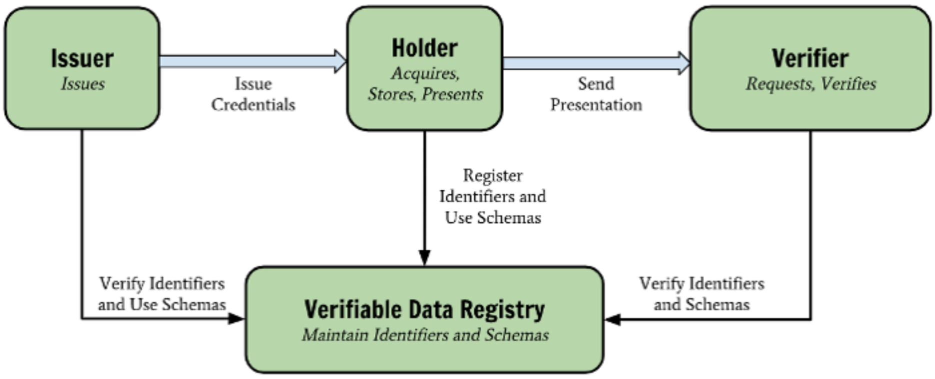

Issuer
Issues
Issue Credentials
Holder
Acquires, Stores, Presents
Send Presentation
Verifier
Requests, Verifies
Register Identifiers and Use Schemas
Verify Identifiers and Use Schemas
Verifiable Data Registry
Maintain Identifiers and Schemas
Verify Identifiers and Schemas

Figure 1. Verifiable Credentials Data Model [32]

The holder acquires the credentials from the issuer and saves the credentials in his/her wallet. The verifier requests the credentials and verifies the credentials by utilizing a verifiable data registry. Indy implements digital hyperledger i.e. blockchain for the verifiable data registry.

3.4. Indy Selection Criteria

To perform the border crossing point checking use case we selected Hyperledger Indy as the blockchain(s) technology platform. Indy was selected based on the following criteria:

1. Hyperledger Indy is designed to support digital identity and self-sovereign concept
2. Hyperledger Indy is provided as open-source code and has an active developer community
3. In Hyperledger Indy, only reference of personal information is saved in the blockchain and real information is in traveller’s wallet in her/his smart phone and issuer’s database.
4. European Blockchain Services Infrastructure (EBSI) utilises similar solutions like Hyperledger Indy

Moreover, blockchain technologies have cybersecurity by design through their inbuild structures, such as the consensus mechanisms, distributed and decentralized nature and built-in secure of blockchain.

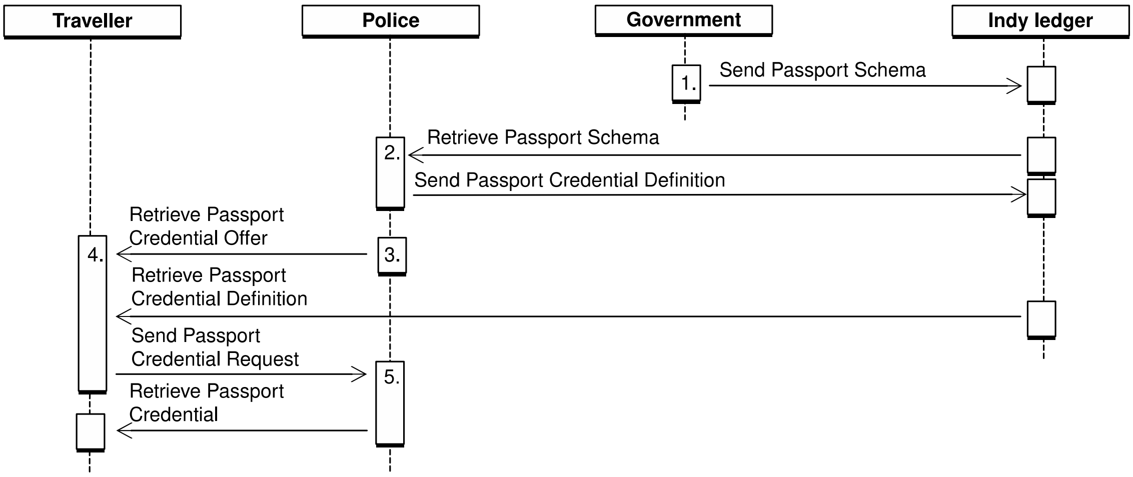
4. BORDER CROSSING POINT CHECKING USE CASES

These use cases handle the traveller’s passport and travel history checking procedures by the border authority. The digital passport is stored into a digital wallet in the traveller’s mobile phone as a verifiable credential. The authority requests the proof of identity, to which the traveller replies by producing passport data with credentials with the help of the blockchain. The credential is automatically verified by the blockchain, and the end result is presented to the border authority. Similarly, the border crossing history of the person, in terms of crossing rejects and acceptances, can be fetched from the authority wallet and verified by the blockchain.

4.1. Passport Checking Use Case

Our passport use case adheres to the Verifiable Credential Data Model described in Figure 1. In the use case, the border authority confirms the validity of the traveller's passport information with the help of the ledger. Thus, the border authority refers to the verifier of the Verifiable Credentials Data Model. The traveller is a holder of the credentials of his/her digital passport. The credentials are provided to him/her by a travel document issuer. The issuer can be for example the national police.

The process can be divided into two parts, first creating and issuing a credential, see Figure 2 and then using that credential, see Figure 3.


Traveller
Police
Government
Indy ledger
1.
Send Passport Schema
2.
Retrieve Passport Schema
Send Passport Credential Definition
Retrieve Passport Credential Offer
4.
3.
Retrieve Passport Credential Definition
Send Passport Credential Request
5.
Retrieve Passport Credential

1. Create a Passport Schema
2. Create a Passport Credential Definition
3. Create a Passport Credential Offer for a traveller
4. Create a Passport Credential Request
5. Create a Passport Credential

Figure 2. Issuing passport credential

4.1.1. Creating and Issuing a Credential

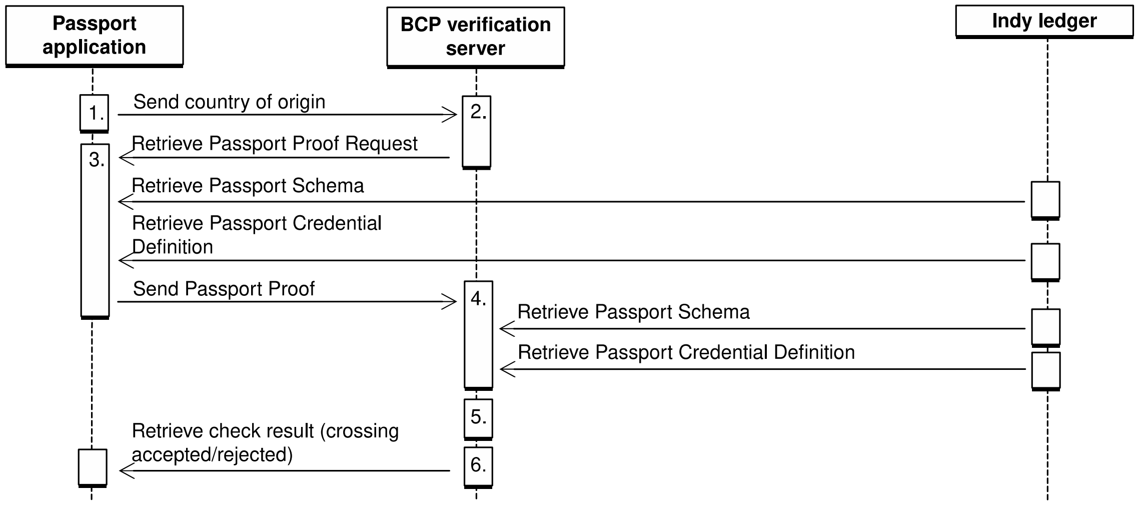
Figure 2 shows the process of traveller obtaining a passport with credential. It also shows actions done earlier to provide this passport service for the citizens of a country.

(1) The format of the digital passport for a country is defined by a governmental body, following international guidelines. The definition is sent to the blockchain as Credential Schema (Ch. 5.4).
(2) An authority to issue passports, here police, fetches the passport schema from the blockchain and creates a Passport Credential Definition (Ch. 5.5) that enables it to issue passport credentials. It sends the definition to the blockchain.
(3) A traveller applies for a new passport from the police (not a blockchain procedure so not shown in the picture). The police starts the issuing (Ch. 5.6) by sending a Credential Offer of a passport to the traveller.

(4) The Credential Offer received by the traveller contains the identifiers of the passport schema and the police credential definition. The traveller retrieves them from the blockchain, builds a Passport Credential Request and sends it to police.
(5) The police creates and signs a Passport Credential and sends it to the traveller. The traveller can now save the Passport Credential to his/her wallet (Ch. 5.3).

4.1.2. Using the Credential

Figure 3 provides a generic outline of the passport border check processes within the implemented use case. The procedure starts when the traveller arrives to the passport checking area of the Border Crossing Point (BCP). A passport verification application is installed in her smart phone. The application may have been configured to detect the area WIFI or the checking procedure may be invoked from an RFID tag or an NFC reader. In our test system, for simplicity, the procedure is started manually.


Passport application
BCP verification server
Indy ledger
1.
Send country of origin
2.
Retrieve Passport Proof Request
3.
Retrieve Passport Schema
Retrieve Passport Credential Definition
Send Passport Proof
4.
Retrieve Passport Schema
Retrieve Passport Credential Definition
5.
Retrieve check result (crossing accepted/rejected)
6.

1. Connect to BCP server
2. Create a Passport Proof Request
3. Create a Passport Proof
4. Verify Passport Proof
5. Check passport travelling history (separate chart)
6. Decide, if crossing is accepted or rejected

Figure 3. Border check process

At the start of the passport verification procedure, the traveller's passport application contacts the BCP verification server. Both the phone and the server have access to the identification blockchain. After the server has accepted the new connection, the passport application (1) sends information of the traveller's country of origin to the server. The BCP server has a list of publicly known credential definition identifiers of the passport issuing authorities of various countries.
(2) The server builds a passport verification proof request, that contains definitions of the passport attributes that must be verified with the credential of the authority in the traveller's country. It sends the request to the phone.
(3) The passport application responses to the request by creating a proof message. From the phone's wallet, it fetches the credential matching the required attributes in the request and adds it to the proof.
(4) The verification server receives the passport verification proof from the phone. It uses the ledger functionality to verify the proof. If the verification succeeds, the passport data included in the proof is verifiably credited by the passport issuing authority (Ch. 5.7).

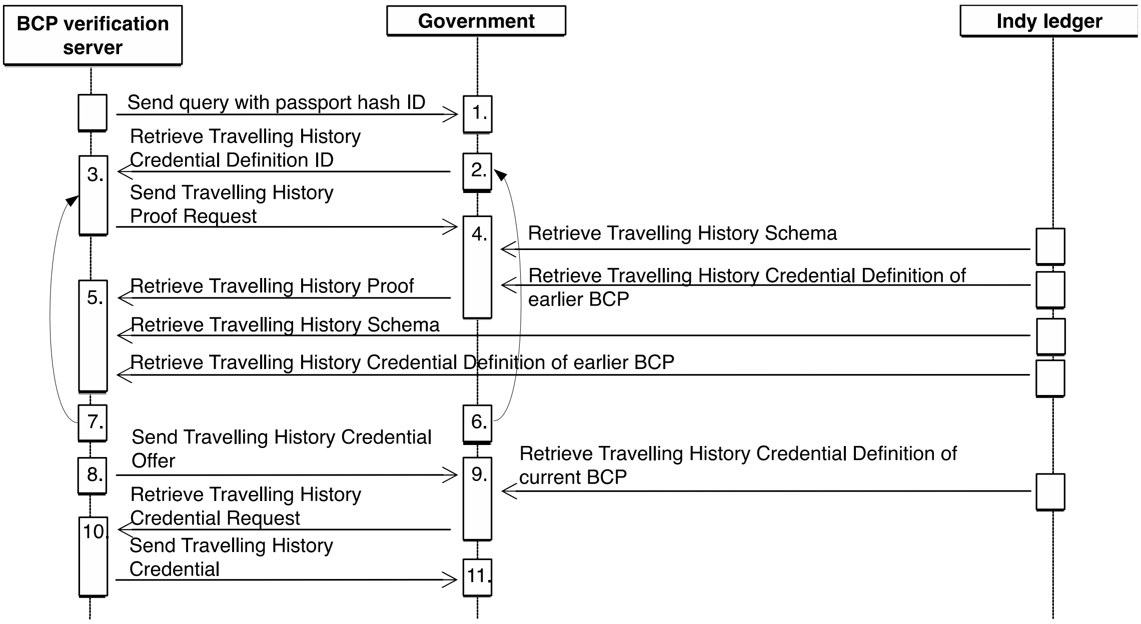
(5) If the passport verification succeeds, the travelling history of the passport can be fetched (Figure 4 and Ch. 4.2).

Other checks not necessarily related to blockchain can then be done automatically. E.g. face recognition or queries to databases, such as national databases, the European Criminal Records Information System on third country nationals holding criminal records (ECRIS-TCN) or Schengen Information System (SIS II).

(6) After all the checks are done, the system can decide, if the traveller’s crossing is accepted or rejected (Ch. 4.3).

4.2. Travelling History Checking Use Case

The BCP server can retrieve and verify earlier border crossings done with this passport and check, if there have been any previous crossing rejects and the reasons to them. The travelling history resides in a government wallet on a central server (or in distributed wallets on several servers, see notes in Ch. 6.3). The BCPs have access to the server(s). They can also send new entries with their own credentials to be included in the wallet. Figure 4 shows the verification and issuing the history records with credentials.


BCP verification server
Government
Indy ledger
Send query with passport hash ID
1.
Retrieve Travelling History Credential Definition ID
2.
3.
Send Travelling History Proof Request
4.
Retrieve Travelling History Schema
Retrieve Travelling History Credential Definition of earlier BCP
Retrieve Travelling History Proof
5.
Retrieve Travelling History Schema
Retrieve Travelling History Credential Definition of earlier BCP
7.
6.
Send Travelling History Credential Offer
Retrieve Travelling History Credential Definition of current BCP
8.
9.
Retrieve Travelling History Credential Request
10.
Send Travelling History Credential
11.

1. Search Indy Wallet for Travelling History credentials with given hash ID
2. Get Travelling History Credential Definition ID of the earlier BCP, that credited the next search result
3. Create a Travelling History Proof Request
4. Create a Travelling History Proof
5. Verify Travelling History Proof
6. Get next search result, if there is one
7. Include the verified travelling history record into check decision and go to wait for the next record, if any
8. After accept/reject crossing, create a new travelling history record and a Travelling History Credential Offer
9. Create a Travelling History Credential Request
10. Create a Travelling History Credential
11. Save credential and travelling history record included in it into Indy Wallet

Figure 4. Travelling history checking process

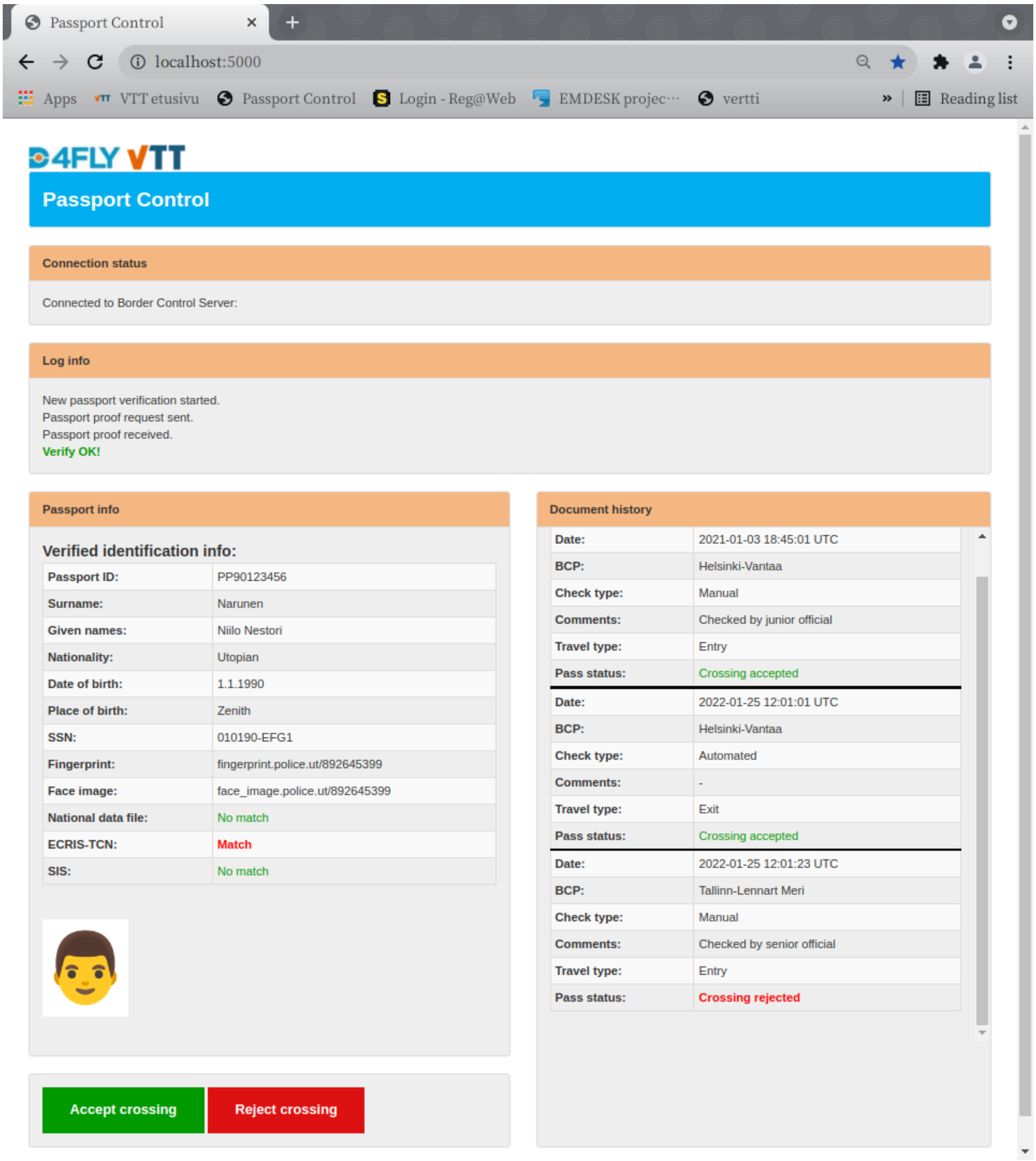
In the travelling history use case, the BCP currently performing the checking process is the verifier or the history record credentials. The holder is the governmental body saving the records. The issuers are the BCPs that have previously performed the verification procedure for the traveller’s passport and sent the checking result as a record with credential to the government wallet. When the current checking is complete, the performing BCP does also that and becomes an issuer.

(1) The BCP server sends a query to the government server, where the history records wallet resides. The query contains a hash value calculated from the verified passport data, as a search key for earlier crossing records.
(2) The government server searches the wallet with the passport hash ID. Each found record contains a credential issued by the BCP that created the record. The current BCP receives the Credential Definition ID of this issuer BCP.
(3) The BCP server builds a travelling history proof request containing attributes, that must be verified with the credential of the issuer BCP. The BCP server sends the request to the government server.
(4) The government creates a proof containing the history record and credential of the earlier BCP. It sends the proof to the current BCP.
(5) The BCP server verifies the history record with the blockchain functionality.
(6-7) The BCP server adds the verified record into list. The BCP and government servers repeat the steps 2-7 until all found records are verified. Then the BCP server adds the record list to its check data.
(8-11) When the BCP has made a decision of accepting or rejecting the crossing, the BCP server builds a travelling history record of the decision, creates a signed credential of it and sends it to the government wallet. The procedure is similar to steps 3-5 of Ch. 4.1.1.

4.3. Crossing Accepted or Rejected

If the passport verification, travelling history and other checks pass, the procedure can be completely automatic, and the traveller receives a confirmation message to the mobile application.

In case of any issues detected in the checking of the traveller, the BCP server has a web GUI with which a border guard can decide, if the crossing can be accepted. Figure 5 illustrated a situation in which a match is found in the ECRIS-TCN database and the previous crossing has been rejected. If the border guard decides to reject crossing, he/she can push the red button and the passport application in traveller's smart phone receives an information message guiding the traveller to proceed to manual check.



Passport Control

Connection status

Connected to Border Control Server:

Log info

New passport verification started.
Passport proof request sent.
Passport proof received.
Verify OK!

Passport info

Verified identification info:

Passport ID:	PP90123456
Surname:	Narunen
Given names:	Niilo Nestori
Nationality:	Utopian
Date of birth:	1.1.1990
Place of birth:	Zenith
SSN:	010190-EFG1
Fingerprint:	fingerprint.police.ut/892645399
Face image:	face_image.police.ut/892645399
National data file:	No match
ECRIS-TCN:	**Match**
SIS:	No match

Document history

Date:	2021-01-03 18:45:01 UTC
BCP:	Helsinki-Vantaa
Check type:	Manual
Comments:	Checked by junior official
Travel type:	Entry
Pass status:	Crossing accepted
Date:	2022-01-25 12:01:01 UTC
BCP:	Helsinki-Vantaa
Check type:	Automated
Comments:	-
Travel type:	Exit
Pass status:	Crossing accepted
Date:	2022-01-25 12:01:23 UTC
BCP:	Tallinn-Lennart Meri
Check type:	Manual
Comments:	Checked by senior official
Travel type:	Entry
Pass status:	**Crossing rejected**

Figure 5. Web GUI of border guard

(Note: For privacy protection, an emoji is used here instead of a real passport picture)

5. Our Indy Implementation

In the passport use case, a traveller's smart phone is used to provide a digital passport, i.e. verifiable credential, which can be automatically verified by the blockchain. The biometric check, if needed, can be made by comparing the biometric measurements (e.g. face, fingerprint, iris) made at the border to the information stored on the wallet. In case of the extra check with border guard intervention, the traveller can control, according to self-sovereign principles, which credential information he or she will provide to the border guard (educational background, employment history, marital status etc.). The final use case also integrates the use case related to travelling history, thus providing extended facilities of digital passport and more information for border checks process.

In our solution, we have implemented the digital passport and a travelling history checking. According to our use case, the border authority would be physically present in the crossing point checking. However, the border crossing point checking event could be further automatized by using the verifiable credentials as digital passport, and automatizing the actual passport checking process with smart contracts.

5.1. Indy Pool and Distributed Ledgers

An Indy pool consists of a network of Indy nodes. It can contain several distributed ledgers. The domain ledger stores the records of identity transactions. In our test case these records consist of created DIDs (Decentralized Identifier), Credential Schemas and Credential Definitions. Each record is associated to exactly one DID [33], whose owner is an Indy agent.

5.2. Roles

Indy have common users and agents, with different roles like Trust Anchor, Trustee and Steward. The travellers are common users, who have a DID and a Wallet. They can query the public identity records from the ledger, but they cannot send data to the ledger. A Trust Anchor or Trustee is an agent, who has a permission to send transactions to the ledger. Trust Anchors can create Credential Schemas, Credential Definitions or DIDs for new agents (Users or Trust Anchors). The DID of a Trust Anchor is of type Verinym. The Verinym is stored on the ledger, so the Trust Anchor is identifiable by others. The BCPs, the police and the government have a Trust Anchor role. A Steward is an operator of the ledger and has all permissions to it, so it can create the initial Trust Anchors. It is a separate agent having its own DID and wallet, in the real world probably also some governmental organization.

5.3. Wallet

The Indy SDK includes a default implementation of a Wallet, a private encrypted storage file for an Indy agent. The wallet stores e.g. the Master Secret, DID and sign keys of the agent, the data with Credentials for which the agent is a holder and the private part of Credential Definitions and Credential Offers for which the agent is an issuer.

Thus, the passport with police credentials is stored into the traveller’s wallet, the private parts of the credential definition and offers into the police wallet. In the test system, the government wallet contains all the travelling history records of all passports - in real world the records must probably be distributed into several wallets. The private part of BCP credentials contained in the records and their definitions reside in the wallets of each BCP involved.

The credentials stored in wallet can be searched with a query language using a logical combinations of e.g. schema and credential definition IDs, issuer DIDs and data values of credential attributes as keys.

5.4. Credential Schema

A Credential Schema defines the attributes, that one Credential Definition can contain. Any Trust Anchor can create a schema and send it to the ledger. After that, it is immutable and available to the ledger users.

The current schema format is a simple JSON array of attribute names. It does not include types, hierarchical structures or encoding required. There are two schemas defined in the test system, both created by the government. The Passport schema defines the fields required in the traveller's passport credentials. The Travelling History schema lists the information to be saved from the passport, or other travel document, check procedure.

5.5. Credential Definition

A Credential Definition can also be created by a Trust Anchor. It is based on a Credential Schema got from the ledger. The public part of the definition is sent to the ledger, signed with the DID of the creator. This announces, that the signee can issue credentials based on the definition. The private part is stored into the creator's wallet.

In our scenario, the police creates a credential definition based on the Passport schema. Each BCP creates its own Credential Definition from Travelling History schema for issuing the credentials to the history records.

5.6. Issuing Credentials

An owner of a Credential Definition can act as an issuer of credentials. It first creates a Credential Offer to a user. The user can then build a Credential Request, which includes this offer, the public part of the Credential Definition got from the ledger and the blinded form of Master Secret, a data residing in the wallet and known only to the user. (According to Indy SDK documentation, Link Secret is currently the preferred term, although Master Secret is still used in the code).

After receiving the Credential Request from the user, the issuer creates a Credential, filling data values according to the attributes of the schema the definition is based. The issuer signs each attribute with the help of the ledger, includes the request with the user's master secret and sends the credential to the user, who becomes now the credential holder.

The passport credentials are issued by the police for the travellers as holders. The BCPs issue their travelling history info credentials for the government as a holder.

The Indy format of credential attributes includes both raw and encoded values. However, the format does not define the encoding type, except that 32-bit integers are encoded by themselves. The test system uses a simple encoding of raw UTF-8 string values to integers.

5.7. Verifying Credentials

When an Indy agent needs a proof, that another agent is a holder of certain credentials, it becomes a verifier and creates a Proof Request. The request contains a list of attributes and predicates, that need to be solved. The attributes may be required to have a credential from a certain definition (identified by their public ID in the ledger). One proof request may include attributes from different definitions. Some attributes may need not to be proved. Predicates may define rules, that certain credentials must meet, although the verifier does not need to know their values.

The verifier sends the Proof Request to the holder, which builds a Proof from the credentials it possesses to meet each attribute in the request. The schemas and credential definitions involved are fetched from the ledger to be included into the proof.

The holder sends the proof as a response to the verifier. The verifier too gets the necessary entities from the ledger and uses the Indy verification functionality to check, if the proof meets all requirements in the request.

The structure of proof requests in the test system corresponds closely to the passport and travelling history info schemas. All attributes in a request must be verified and they are all from the same credential definition.

6. TEST AND DEMO SETUP

In our test and demo system, we utilized the software provided by two Hyperledger GitHub repositories, Indy Node [34] and Indy SDK [35]. The Python code of Indy Node implements the functionality of validator and observer nodes, that form the Indy blockchain distributed ledger. Indy SDK contains Rust language code, that builds into a C-callable Indy client library. The library enables an application to create the various Indy components and to communicate with the blockchain nodes pool. Library wrappers in several programming languages are provided. We chose the Python wrapper for our implementation. The SDK contains also a Docker file, with which a test pool image with four Indy nodes can be built.

The test system consists of two Indy agents communicating with the blockchain and with each other. One agent resides in the mobile phone of a traveller, another in the server of a Border Control Point. However, although an Android version of the Indy library existed, it was still in experimental stage according to Indy documentation. Because of this and because the Indy development with Python was better documented, we chose to implement the traveller's Indy functionality on a separate Python server. The test system architecture consists therefore of two servers, with which the traveller's mobile communicates.

6.1. Server Implementation

Both the BCP and Traveller server-side code run in one Python process in an Ubuntu Linux machine. The Indy pool of four nodes run inside a Docker image in the same machine. Indy libraries are available as APT packages for Ubuntu 16.04 and 18.04. The latter seems to install and function without problems in Ubuntu 20.04 too.

A Python 3 application contains the BCP and traveller Indy functionality. It uses the Indy library via Indy Python package. For the BCP server, a web GUI using the Flask web framework is implemented.

6.2. Mobile Implementation

The passport application for the traveller's mobile is implemented in Android Java. In the test system, it can be run in an Android phone or in an Android SDK emulator.

On the start of the passport check, the application makes a TCP connection to the traveller's server, which contains the traveller's wallet and has access to the blockchain pool. Then the application connects also to the BCP server to start the check procedure. The application forwards the proof request from the BCP to the traveller's server, which creates a proof response containing the credited passport info from its wallet. The server sends the proof to the mobile application, which forwards it to BCP server. Finally, the BCP server sends a crossing confirmation or reject to the application.

6.3. Test System Restrictions

In a real world, the blockchain would naturally be distributed to several nodes across the network. Likewise, there would be several server machines accessing the nodes and handling the Indy transactions of various agents.

An obvious difference to the test system is the architecture of the government wallet. Instead of one centralized entity, a hierarchical search system over distributed wallets on distributed servers should be designed.

Each BCP would probably have an own server, that contained its wallet. It would also have access to the government wallet system to save and fetch travelling history records.

One alternative could be, that each BCP wallet would contain the travelling histories of its own entries and exits. The wallet would be added to distributed search system for queries of other BCPs.

The TCP connections between the application and the servers are unencrypted. The real world-connections must of course use encryption, although this too is outside the blockchain functionality.

7. Discussion and Future Directions

The implementation of the basic functionality using the Indy libraries and tools was fairly easy. The API of the Indy library and its Python wrapper is not trivial, but the Getting Started Guide and the Python example code from the SDK enabled a quick implementation. Functionality covered in examples included e.g. connection to the pool, creation of DIDs, wallets, schemas and credential definitions, issuing a credential and verifying it. Also, it was very straightforward to build and start the Indy test pool Docker image from the SDK, to be run on host Ubuntu machine. Beyond this basic functionality, the Indy documentation was less thorough. It was also scattered to various websites, which made the search more difficult. It was also unclear, which described features were already implemented in current version of Indy, which were only future plans, or which were even superseded by new features. E.g. DID documents and service endpoints could have been usable in our solution, but although there was some documentation of them and even some endpoint-related functions in Indy API code, it turned out they were not yet implemented. When the documentation was lacking, the Indy functionality could of course be studied from the Python and Rust code itself - the advantage of open source software. Although this may sometimes be time-consuming.

The current Indy system has no in-built search for multiple wallets. The distributed search system for document history described in Ch. 6.3 must therefore be implemented with tools outside Indy. According to self-sovereign data principles, personal information should not be needed to be shown more than it is required. In our solution based on verifiable credentials, it could be an option that the private information is shown partially, or optionally it could be hidden totally from the border control authorities. There is not necessarily a need to show some or any private information to border controllers, unless there are regulations demanding it. In the latter case what is needed is only confirmation from the system that the authorization is succeeded and the trust from the border control that the system is functioning properly. In other words, the border control process we have described preserves user privacy.

Our implementation uses a WiFi-wireless connection for data exchange between the smart phone and the blockchain infrastructure. Other methods Bluetooth, RFID, QR-code etc. for exchanging data can be considered in future implementation. Furthermore, in order to associate the smart phone with the user, some biometric information is required from the user to verify the owner of the phone. In prospective border control systems, the quality and amount of personal and biometric information that is needed from the user in the border check process needs to be standardised and refined in order to maintain privacy and self-sovereignty. Regarding automatisation level, it is possible that almost a fully automatised border control process will be enabled in the future. This imposes that the human role and the automated parts in the border control process should be carefully reconsidered, especially in terms of what is the trust based in the system.

In this research, the scope of our study is in SSI implementation that uses blockchain technologies, and is applied in the border control context. Evaluation of the end-to-end security of the system, including secure wireless communication between mobile application and servers as well as data protection in servers, requires planning and implementing the system in real life. However, the communication in wireless systems and the protection of servers can use common data encryption (like AES and RSA) or new post quantum cryptography solutions when these will be have standardized.

Regarding the DLT as a technology, blockchains in general have clear advantages when applied in the border control environment to enhance the self-sovereign approach. There are open platforms available for developers (such as Indy or Ethereum), and the distributed approach minimizes some security threats such as single point of failures that apply traditional centralized systems. In addition, smart contracts could be used to automatize functionalities. As discussed earlier, the whole border control process could actually be automated without or with minimal need of human interventions. Self-sovereign data handling, overall data and system management, as well as the technology behind the blockchain are quite mature. However, the technology may still not necessarily be familiar to all. On top of that, issues like scalability and system validation need more study. An important aspect to consider in future work is the understandability or explainability of the solution’s back-end side, as decentralised systems or the blockchain technology itself still are somewhat unfamiliar in the border security context. Border guards need a sufficient overview of the mechanisms and techniques upon which the blockchain solutions are built to improve the understanding and trust in the systems. Increased automation in the border control environment requires high level of trust in the adopted solutions, and also in the results presented to the border guards through a graphical user interface. High processing speed (potentially less than 1 second) makes any manual overseeing of individual processing steps impractical and vain also from the perspective of the traveller. As the level of automatization increases in the border control environment, more standardization work is required, just as discussed regarding the use of biometric data.

From the security point of view, if implemented correctly, the blockchain technology may improve overall security via its distributed approach and cryptographic assurance of the blockchain concept. However, to utilise the blockchain technology in the context of border control on a larger scale, there is a need to create more comprehensive blockchain infrastructure in terms of co-operation between countries and the related actors. This requires a significant amount of political effort, co-operation and decision-making among the domains related to border-crossing and different authorities.

The concept of digital identity in the form of SSI is important not only in border control, but also in other citizens interactions, where the citizens must prove their identity (e.g. in interaction with authorities, health care, educational systems and monetary institutions). At the European level, the initiative of EBSI acts as a support and acceleration for the digital identity development, by leveraging blockchain in the creation of cross-border services for public administrations and their ecosystems. Regarding further development and appliance of SSI within the border control environment, the SSI use case demonstration provided by EBSI will be a good starting point.

8. CONCLUSIONS

In the border control domain, the next evolution step is the development of eID, including self-sovereign aspects. In this study, we developed a small scale demo to present mentioned evolution by using Hyperledger Indy distributed ledger technology (DLT). Our solution is based on verifiable credentials, where the private information can be partially disclosed, or totally hidden from the border control authorities. The demo implements two border check use cases, where one provides “digital passport” and the second "travelling history" information. Our main aim was to investigate the suitability of DLT in providing data self-sovereign applicability with eID within border checks, and examine the benefits and disadvantages of the technology.

Our solution uses WiFi-connection for data exchange between the smart phone and the blockchain infrastructure. In future work, other connection methods can be studied. Travelling history checking use case requiring distributed search through distributed wallets and services will need a new design on top of our solution.

In future, the use of automatised border checks will be increasing and in this, the relationship between the border control personnel and the automation should be carefully defined. Regulations such as GDPR will keep enforcing to disclose a minimal amount of private information to border controllers, but for solutions such as ours, some biometric information is still needed. Research and standardisation effort is needed in optimizing the use of biometric data in border inspections, especially when advancing the approach of SSI management.

From the technology point of view, DLT brings about intrinsic security based on blockchain technology supporting concepts of digital and self-sovereign identity. However, for some fields like border control domain, it is still rather new technology and requires more documentation, standardisation and demonstrations. Education, and solutions to issues like scalability and validation will need to be advanced. In EU, a good embodiment for future innovations is offered by EBSI.

REFERENCES

[1] F. European Agency for the Management of Operational Cooperation at the External Borders of the Member States of the European Union, (2012) “Best practice operational guidelines for automated border control (abc) systems,” Tech. Rep., 2012.

[2] “Impact assessment of the covid-19 outbreak on international tourism,” (2022), accessed: 2022-8-2. [Online]. Available: https://www.unwto.org/impact-assessment-of-the-covid-19-outbreak-on-international-tourism

[3] E. Border and F. Coast Guard Agency, Eds. (2020), *International Conference on Biometrics for Borders - Morphing and Morphing Attack Detection Methods.* Warsaw, Poland: European Border and Coast Guard Agency, Frontex.

[4] S. European Union Agency for the Operational Management of Large-Scale IT Systems in the Area of Freedom and e.-L. Justice, (2021) “Contactless travel in post-covid times: Enhancing the eu security ecosystem,” Tech. Rep.

[5] “New abc egates: smaller footprint, modular design and faster passenger processing,” (2022), accessed: 2022-8-2. [Online]. Available: https://www.thalesgroup.com/en/markets/digital-identity-and-security/government/eborder/egates

[6] "The digital travel credential (dtc), a step towards a simplified travel experience," (2022), accessed: 2022-8-2. [Online]. Available: https://www.ingroupe.com/en/digital-travel-credential-dtc-simplified-travel-experience

[7] I. International Civil Aviation Organization, (2020) "Guiding core principles for the development of digital travel credential dtc," Tech. Rep.

[8] U. von der Leyen, "State of the union address. (2020)" Presented as the 2020 State of the Union Address, Brussels, Belgium.

[9] E. Commission, "Communication from the commission to the european parliament, the council, the european economic and social committee and the committee of the regions 2030 digital compass: the european way for the digital decade," (2021). accessed: 2022-8-2. [Online]. Available: https://ec.europa.eu/info/sites/default/files/communication-digital-compass-2030_en.pdf

[10] E. Commission, "Commission proposes a trusted and secure digital identity for all europeans," (2021). accessed: 2022-8-2. [Online]. Available: https://ec.europa.eu/commission/presscorner/detail/en/IP_21_2663

[11] E. Commission, "Commission recommendation (eu) 2021/946 of 3 june 2021 on a common union toolbox for a coordinated approach towards a european digital identity framework," (2021).accessed: 2022-8-2 [Online]. Available: https://www.stradalex.com/fr/sl_src_publ_leg_eur_jo/toc/leg_eur_jo_3_20210614_210/doc/ojeu_2021.210.01.0051.01

[12] "The european digital identity framework," (2022), accessed: 2022-6-2. [Online]. Available: https://www.worldbank.org/content/dam/photos/1440x300/2022/feb/eID_WB_ presentation_BS.pdf

[13] T. E. PARLIAMENT and T. C. O. T. E. UNION, "Regulation (eu) 2016/679 of the european parliament and of the council of 27 april 2016 on the protection of natural persons with regard to the processing of personal data and on the free movement of such data, and repealing directive 95/46/ec (general data protection regulation)," (2016). accessed: 2022-8-2. [Online]. Available: https://eur-lex.europa.eu/eli/reg/2016/679/oj

[14] T. E. PARLIAMENT and T. C. O. T. E. UNION, "Regulation (eu) no 910/2014 of the european parliament and of the council of 23 july 2014 on electronic identification and trust services for electronic transactions in the internal market and repealing directive 1999/93/ec," (2014). accessed: 2022-8-2. [Online]. Available: http://data.europa.eu/eli/reg/2014/910/oj

[15] H.Funke, "Digital and mobile identities, (2020)" in *Open Identity Summit 2020,* H.Roßnagel,C.H. Schunck, S. Mödersheim, and D. Hühnlein, Eds. Bonn: Gesellschaft für Informatik e.V., pp. 27–33.

[16] A. J. Zwitter, O. J. Gstrein, and E. Yap, (2020) "Digital identity and the blockchain: Universal identity management and the concept of the "self-sovereign" individual," *Frontiers in Blockchain*, vol. 3, 2020. accessed: 2022-8-2. [Online]. Available:https://www.frontiersin.org/article/10.3389/fbloc.2020.00026

[17] "Digital identity: Leveraging the ssi concept to build trust," accessed: 2022-8-2. [Online]. Available: https://www.enisa.europa.eu/publications/digital-identity-leveraging-the-ssi-concept-to-build-trust

[18] "Self-sovereign identity use case, (2022) " accessed: 2022-8-2. [Online]. Available: https://ec.europa.eu/digital-building-blocks/wikis/display/EBSI/Use+cases

[19] D. Pinto Lopes, P. Rita, and H. Treiblmaier, (2021) "The impact of blockchain on the aviation industry: Findings from a qualitative study," *Research in Transportation Business & Management*, vol. 41, p. 100669. accessed: 2022-8-2. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S2210539521000523

[20] R. W. A. B. P. A. R. Banerji, Diptiman, (2021), *Application Potential of Blockchain Technologies in the Travel and Tourism Industry*. Cham: Springer International Publishing, pp. 289–299. accessed: 2022-8-2. [Online]. Available: https://doi.org/10.1007/978-3-030-65691-1_19

[21] "Known traveller digital identity, specification guidance," (2020), accessed: 2022-8-2. [Online]. Available https://www3.weforum.org/docs/WEF_KTDI_Specifications_Guidance_2020.pdf

[22] "Accelerating the transition to digital credentials for travel: Lessons from ktdi – a public-private collaboration for secure and seamless travel," (2021), accessed: 2022-8-2. [Online]. Available: https://www3.weforum.org/docs/WEF_Accelerating_the_Transition_to_Digital_Credentials_for_Travel_KTDI_Playbook_2021.pdf

[23] P. Gugler, M. Alburai, and L. Stalder, (2021) "Smart city strategy of dubai".

[24] M. S. Khan, M. Woo, K. Nam, and P. K. Chathoth (2017), "Smart city and smart tourism: A case of dubai," *Sustainability*, vol. 9, no. 12. accessed: 2022-8-2. [Online]. Available: https://www.mdpi.com/2071-1050/9/12/2279

[25] A. A. Shoul, (2017) “Now, smartphone is your passport in dubai,” *Gulfnews*, accessed: 2022-8-2. [Online]. Available: http://gulfnews.com/news/uae/emergencies/now-smartphone-is- your-passport-in-dubai-1.2040149

[26] M. Htet, P. T. Yee, and J. R. Rajasekera, (2020) “Blockchain based digital identity management system: A case study of myanmar,” in *2020 International Conference on Advanced Information Technologies (ICAIT)*, pp. 42–47.

[27] D. Bissessar, M. Hezaveh, F. Alshammari, and C. Adams, (2018) “Mobile travel credentials,” in *International Symposium on Foundations and Practice of Security*. Springer, pp. 46–58.

[28] V. Patel, (2018) “Airport passenger processing technology: a biometric airport journey”.

[29] C. Allen, (2016) “The path to self-sovereign identity,” accessed: 2022-8-2. [Online]. Available: http://www.lifewithalacrity.com/2016/04/the-path-to-self-soverereign-identity.html

[30] D. Tobin, A. Reed, (2017) “The inevitable rise of self-sovereign identity”. accessed: 2022-8-2. [Online]. Available: https://sovrin.org/wp-content/uploads/2017/06/The-Inevitable-Rise-of- Self-Sovereign-Identity.pdf

[31] U. Feige, A. Fiat, and A. Shamir, (1988) “Zero-knowledge proofs of identity,” *Journal of cryptology*, vol. 1, no. 2, pp. 77–94.

[32] “Verifiable credentials data model v1.1,” (2021), accessed: 2022-8-2. [Online]. Available:https://www.w3.org/TR/vc-data-model/

[33] “Indy walkthrough, indy sdk github site,” (2022) , accessed: 2022-8-2. [Online]. Available: https://github.com/hyperledger/indy-sdk/blob/master/docs/getting-started/indy-walkthrough.md

[34] “Indy Node Github repository (2019)” ,” accessed: 2022-8-2. [Online]. Available: https://github.com/hyperledger/indy-node

[35] “Indy SKD Github repository,” (2021)”, accessed: 2022-8-2. [Online]. Available: https://github.com/hyperledger/indy-sdk

AUTHORS

Pekka Koskela received the D.Sc. degree in 2018. Over 20 years he has been worked in several research projects both researcher and project leader. Currently he is studying among others the exploitation of quantum, homomorphic and digital ledger technologies.

Anni Karinsalo has been working as a Senior Scientist at VTT’s Applied Cryptography team since 2004. She has experience from several fields of cybersecurity from her work as a project leader and researcher, such as post-quantum cryptography, blockchains, security metrics and critical infrastructure security.

Jori Paananen received his MSc degree (telecommunications) from Helsinki University of Technology (HUT) in 1983. Since then he has worked at VTT Technical Research Centre of Finland on research projects in various areas of software development. Currently he is Senior Research Scientist in the Connectivity Networks team.

Laura Salmela works as a Project Manager and a Research Scientist in the Critical Cyber-Physical Systems team at VTT. She holds a Master’s Degree in Social Sciences (International Relations) from the International School of Social Sciences at the University of Tampere. Her core competencies are in-depth knowledge of security-critical domain.