Verifiable Contracting

A use case for onboarding and contract offering in financial services with eIDAS and Verifiable Credentials

Sérgio Manuel Nóbrega Gonçalves[2][0000−0002−7818−4757], Alessandro Tomasi[1][0000−0002−3518−9400] Andrea Bisegna[1,3][0000−0002−8055−5262], Giulio Pellizzari[2][0000−0001−6455−780X], and Silvio Ranise[1][0000−0001−7269−9285]

[1] Security & Trust, FBK, Trento (Italy) {a.bisegna,ranise,altomasi}@fbk.eu
[2] University of Trento, Trento (Italy)
{giulio.pellizzari,sm.nobregagoncalves}@studenti.unitn.it
[3] DIBRIS, University of Genoa, Genoa (Italy)

Abstract. We investigate the combined use of eIDAS-based electronic identity and Verifiable Credentials for remote onboarding and contracting, and provide a proof-of-concept implementation based on SAML authentication. The main non-trivial value derived from this proposal is a higher degree of assurance in the contract offering phase for the Contracting Service Provider.

Keywords: Verifiable Credentials · Digital Identity Proofing · Digital Contracting

1 Introduction

From the point of view of a Service Provider (SP), offering a contract to a remote applicant can be a risky proposition. Reliably establishing firstly that someone not physically present really is who they claim to be, and secondly that the details they have provided to enter into a legally binding agreement with the SP are correct, is no trivial task. There are long established procedures to address these problems when the person is physically present, usually involving a form of photographic ID and any number of proofs of other details, such as utility bills to prove current address and/or bank statements to prove account numbers. In the case of remote applicants, however, digital solutions for identity proofing and remote contracting are still a work in progress. In this paper, our contribution is a proof-of-concept to test the combination of two recent and emerging technologies: electronic identity cards and verifiable credentials.

We consider a Contracting Service Provider (CSP) wishing to enter a legally binding contract with an applicant before providing their services. Important examples include utilities and telecoms. In establishing a legally binding contract, CSPs commonly incur costs due to fraud and erroneously entered information. Concretely, a utility billing the wrong bank account, or having to enter a legal dispute over information entered by an applicant during a past contracting

phase, will incur legal costs and delays. In this work we focus on the initial offering phase, in which a CSP wishes to have a high degree of assurance that the information being entered in the contract is correct before offering it to the applicant; our goal is to determine whether the combination of two innovative technologies can assist in this process.

As a concrete minimal example, we consider the case of a utility CSP requiring (a) an applicant's personal information and (b) an applicant's bank account number (IBAN). The eIDAS [9] framework is designed to enable a public service infrastructure for secure and remote identity proofing[4], and the potential of eIDAS-based eID for strong customer authentication in the banking sector is well-known - see for instance [6]. The Verifiable Credentials W3C recommendation [15] is designed to enable the sharing of verifiable claims about subjects with cryptographic proofs of integrity and authenticity.

In this paper we examine how these two frameworks could be usefully combined in order to enable secure remote contracting. To the best of our knowledge, this is the first PoC combining the two technologies. The main non-trivial value derived from this proposal is a higher degree of assurance in the contract offering phase for the CSP. Considering the novelty particularly of the VC recommendation, our objective was first and foremost to test the practical feasibility of the idea; we leave a proper security assessment to future work.

In Section 2 we describe the use case and a scenario we propose to address it. In Section 3 we briefly summarize some of the relevant aspects of the technologies in our proposal. In Section 4 we describe our proof-of-concept implementation. Finally, in Section 5 we evaluate our findings.

2 Use case: contract offering

We consider the case of a utility CSP requiring (a) an applicant's personal information and (b) an applicant's International Bank Account Number (IBAN) in order to offer them a contract for services. In the case of an unknown applicant, this information will be considered Claims by the applicant and which the CSP will have to either verify or accept at their own risk. In order to mitigate against fraud, the CSP would prefer a high level of assurance in these claims, and a means of verifying that: the claims are correct and valid, the applicant at the time of offering is the same as the claim subject, and the issuer is a trusted party.

2.1 Proposed use case scenario

Concretely, our proposal is to consider the IBAN to be an attribute of a Subject, and to have the Account Servicing Payment Service Provider (ASPSP) issue a

[4] Identity proofing is the process of establishing that an unknown applicant really is who they claim to be, and is performed during customer onboarding (e.g. opening a new bank account); after onboarding, accounts are associated with an authenticator, and subsequently authentication is required for a remote claimant to access an enrolled identity's resources (e.g. online banking). See [11].

Verifiable Credential to that effect. A rough component diagram of our proposal is shown in Figure 1.

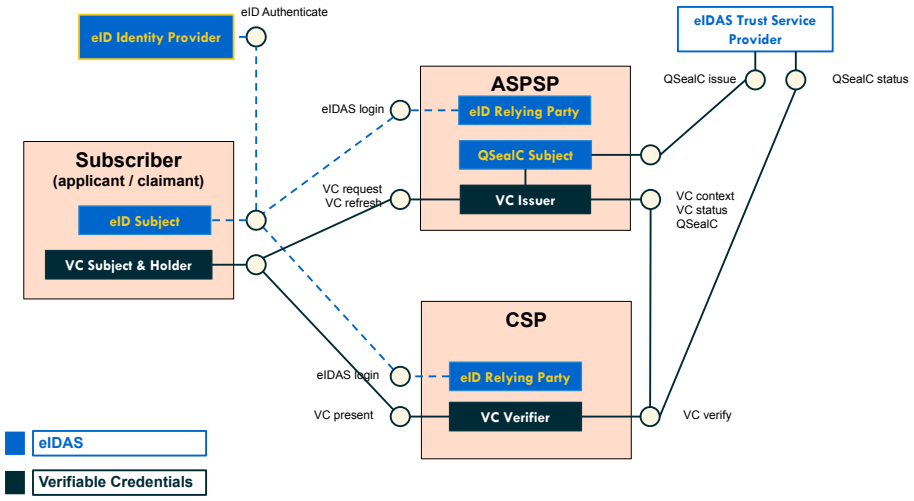

Fig. 1: Entities involved in the proposal and their roles under the two main trust frameworks - eIDAS and VC.

From the perspective of identity management in the cybersecurity context of, e.g. SP 800-63B [11], the required information about the applicant can be considered as attributes of an identity, i.e. claims made about a Subject by an Issuer or Identity Provider (IDP) with some associated proofs of authenticity.

The Verifiable Credentials [15] W3C recommendation is designed specifically to enable the sharing of verifiable claims about subjects with cryptographic proofs of integrity and authenticity.

In order to accomplish this, Issuer (ASPSP) and Verifier (CSP) of the VC can identify the Subject by their eIDAS unique identifier (See Section eIDAS-based eID for identity proofing). The eIDAS framework is designed to enable a public service infrastructure for secure and remote identity proofing; the potential of eIDAS-based eID for strong customer authentication in the banking sector is well-known - see for instance [6]. For instance, one of the eIDAS-notified schemes is the Italian eID card, CIE 3.0, and its use as a means of identity proofing during remote onboarding is already explicitly permitted Italy Bankitalia AML regulations, which state that authentication through an eIDAS-compliant scheme is sufficient to perform due diligence for the specific step of identity proofing, even without the physical presence of the applicant ([2] part 2 section III comma 2).

Using eIDAS, a citizen with an eID is the Subject of identity assertions by their national IDP. The Subject acting as a Subscriber first applies for an account at an Account Servicing Payment Service Provider (ASPSP, in the sense of PSD2), then receives a contracting offer from a Contracting Service Provider (e.g. utilities or telecoms).

Using Verifiable Credentials, the ASPSP issues a verifiable claim tying an account number to the Subject; the Subject holds the claim in their wallet, and presents the claim to the CSP in order to receive a contract offer.

At a high level, the proposed steps would be the following:

1. Requester, in possession of eID, requests a new account with Account Service Provider (ASP)
2. ASP performs automated remote identity proofing with eID through 'login with eIDAS'
3. Bank issues a VC with the requester's eID PersonIdentifier' as subject and the new IBAN as attribute
4. Requester requests a contract offer from Service Provider (SP)
5. SP performs automated remote identity proofing with eID through 'login with eIDAS'
6. SP verifies VC of type IBAN and matches VC subject attribute with eIDAS 'PersonIdentifier'

3 Background: eIDAS and SAML SSO

3.1 eIDAS-based eID for identity proofing

eIDAS allows Relying Parties (RP) to receive SAML assertions [14] on a core attribute set [7] of eID bearers from the eIDAS attribute profile. For natural persons, these are summarized in Table 1.

Table 1: eIDAS attributes for natural persons ([7]).

Mandatory	Optional
Current Family Name	First Names at Birth
Current First Names	Family Name at Birth
Date of Birth	Place of Birth
Unique Identifier	Current Address
	Gender

3.2 SAML SSO

In this paper we consider the SAML 2.0 Web Browser SSO Profile [14] (SAML SSO) since the concrete eID scheme we have in mind is based on a SAML 2.0

IDP [12], and the web browser profile fits our use case (Section 2) and our implementation (Section 4). Fully mobile and hybrid scenarios are also considered in the documentation but beyond the scope of the present proof of concept.

Three entities are involved: a Client (C), an Identity Provider (IDP), and a Service Provider (SP). C is a web browser with which a user interacts; the user's goal is to have access to a service or a resource provided by the SP. IDP authenticates C and issues authentication assertions that are trusted by SP - the SSO trust relationship is depicted with a handshake icon in Figure 2. SP uses the assertions generated by the IDP to decide on C's entitlement to the requested service or resource.

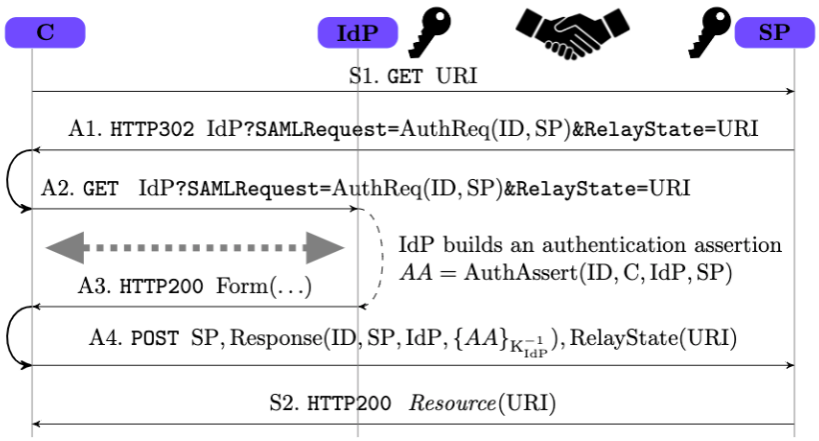

Fig. 2: Message Sequence Chart (MSC) of the SAML SSO protocol [1].

Figure 2 shows a Message Sequence Chart (MSC)[5] of the main steps of the SAML SSO protocol, which we can briefly describe as follows:

S1 C asks SP to provide the resource at URI.

A1-2 SP sends C an HTTP redirect response (status code `302`) for IDP, containing an authentication request AuthReq(ID,SP), where ID is a (pseudo-)randomly generated string uniquely identifying the request (steps A1 and A2). A frequent implementation choice is to use the `RelayState` field to carry the original URI that C has requested (see [14]).

↔ IDP challenges C to provide valid credentials (dotted double arrows in the figure): this is not specified in the standard of the SAML SSO in order to accommodate any authentication process offered by IDP.

[5] Each vertical line in an MSC represents an entity, and horizontal arrows represent messages from one component to another. Identity manangement protocols are often expressed as MSC to identify any flaws.

A3-4 If the authentication succeeds, IDP builds an authentication assertion as the tuple AA = AuthAssert(ID, C, IDP, SP) and embeds it in a response message Resp = Response(ID, SP, IDP, $\{AA\}_{K_{IDP}^{-1}}$) where $\{AA\}_{K_{IDP}^{-1}}$ is the assertion signed with IDP's private key (the key icon in Figure 2). IDP places Resp and the value of RelayState received from SP into an HTML form and sends the result back to C in an HTTP response (step A3) together with some script that automatically posts the form to SP (step A4).

S2 Finally, the SP sends C an accepted HTTP response (status code `200`) containing the requested resource.

3.3 eIDAS-compliant certificates and PSD2

eIDAS-compliant Qualified Certificates conforming to ETSI TS 119 495 [8] are the standard for PSD2 API for both authentication (Qualified Website Authentication Certificates - QWAC) and non-repudiation / content commitment (Qualified Electronic Seal Certificates - QSealC).

The Berlin group access-to-account implementation guidelines [4] require mutual authentication of TPP and ASPSP using eIDAS- and RTS-compliant Qualified Certificates, which must include all the roles for which the TPP is authorized. Open Banking Europe[6] maintains a list of Qualified Trust Service Providers issuing PSD2-compliant Qualified Certificates.

4 Scenario and implementation

The entities involved in the scenario and their roles are:

1. eID holder - Subject of an IDP-issued eIDAS-based eID document
2. eID IDP
3. eID OCSP responder
4. ASPSP - Relying Party to IDP under eIDAS, QWAC and QSealC Subject under eIDAS, Issuer of VC
5. CSP - Relying Party to IDP under eIDAS, Verifier of VC

Our eID subjects are authenticated to SPs through an X.509 certificate, designed to resemble the basic elements of the Italian eID certificate specifications [13]. In particular, the Subject commonName field contains an unique identifier of the person independent of the individual certificate or document, the only allowed key usage is authentication (digital signature), and extended key usage is client authentication.

All the servers (entities 2-5) are developed using NodeJS, and their services have been configured to work over a secure communication channel (HTTPS) to protect them from man-in-the-middle attacks. Servers have two separate certificates, one for server authentication and one for non-repudiation. In general, these could be issued by any authorized CA; in our specific use case, we think

[6] https://www.openbankingeurope.eu/qtsps-and-eidas/

it plausible that these would be Qualified Website Authentication (QWAC) and Qualified Seal Certificates (QSealC), respectively.

Our sample implementation is concerned mainly with the Service Provider part of the architecture supporting authentication and verifiable credential issuing and verificaition to support the use case. Our proof-of-concept implementation is available on github[7]. We give a high-level description here and refer the interested reader to the repository for implementation details.

4.1 SAML

The two SPs (entities 4,5) implement SAML through the passport-saml module. The IDP uses the saml-idp module.

After receiving the SAMLRequest from the SP, the IDP verifies if together with the authentication request the client has also provided the certificate. If that is the case, a verification process starts. The IDP checks if the client certificate has been signed by the Certification Authority (CA) it expects, whether it has expired, and finally whether it has been revoked. The latter operation is achieved by an API call to the OCSP service, which exposes an API that accepts as input a certificate, checks if its serial number belongs to the list of revoked ones and returns 'good', 'revoked', or 'unknown' accordingly.

If all these checks are successful and the user grants access to their data to the SP, the SAMLResponse is generated and sent back to the SP, which parses it and shows the contained attributes.

The SAML implementation has been designed with a view towards integration with our container-based identity management training environment, Micro-Id-Gym [5].

4.2 Verifiable Credentials

Verifiable Credentials (VC) allow Issuers to issue Claims - signed statements about Subjects. Issuers are identified by a URI, Subjects may be identified by a URI or a set of attributes.

We highlight the following steps taken to adapt the VC data model [15] to our use case, and we note that the issuer has to provide information about itself and the credentials it has issued via specific endpoints, in a manner not unlike an identity provider. The endpoints are to be taken as following the issuers domain, `https://<issuer_host>`[8]. An example of an issued VC is shown in Listing A (Section A).

verification The VC has an embedded proof property constructed as a digital signature with the issuers non-repudiation private key, corresponding to their trust provider-issued non-repudiation certificate. The public key can be obtained from the controller document at the `/issuer` endpoint.

[7] https://github.com/stfbk/vc-saml-node.

[8] In our simple nodejs-based proof-of-concept implementation, this is `localhost` followed by a port identifying the service provider.

credential type and context We needed to introduce the subject attributes of eIDAS unique identifier and IBAN in the VC issued by the ASPSP. This was done by defining a custom context, which the issuer makes available through an `/credential/iban` endpoint.

issuance, expiration, and status Issuance and expiration dates are added, and a `/credential/status` endpoint can be called to check whether the VC has been revoked.

5 Lessons learned and conclusion

Added value The main non-trivial value derived from this proposal is a higher degree of assurance in the contract offering phase for the Contracting Service Provider. While there are some costs and technical know-how required in becoming a Service Provider under eIDAS, these are predictable costs and expected to be quite small, as opposed to costs incurred as a consequence of fraud, litigation against repudiation, and plain errors due to manually entered data.

The ability to perform identity proofing remotely is of course highly valuable, but on its own it is enabled by eIDAS as an explicit design goal, and is not new or specific to this proposal. At the same time, for a financial use case it is extremely plausible to use an eIDAS login as a starting point since it strongly contributes to an ASPSP's AML compliance. The addition of Verifiable Credentials based on eIDAS is a synergy expected to enable an ecosystem of high-assurance contracting services.

ASPSP as VC Issuers The solutions adopted by financial services providers often form the gold standard for identity proofing and authentication, and in some cases banks act as identity providers themselves (e.g. BankID [3]). It is not unreasonable to assume that financial institutions would be willing to offer VC issuing services; the set-up involved is in some ways less onerous, in the sense that they do not require federation between Issuer and Verifier, and the Subject is responsible for their sharing.

With reference to the API commonly proposed to comply with PSD2, VCs also appear more adequate for sharing long-term information that may be considered an attribute of the subject, as opposed to live information about their ASPSP-managed account, such as availability of funds and initiation of payments etc. In our proposal, a contracting SP does not have to take the subject's identity attributes and request information about a related IBAN through a PISP; the SP can immediately match the subject's authentication to the subject of the VC and only has to verify the validity of the VC itself.

Contract signing We note that an important piece we have not covered in this proposal is how to close the contracting phase with an electronic signature carrying adequate legal weight. Notably, there exist available solutions based on eID cards, such as `https://firmoconcie.it/` for the Italian CIE 3.0. Alternatively, just as legal persons can apply for a QSealC, natural persons can also be issued

Qualified Certificates. In any case, the signature process would require a careful study of client-side issues such as the informed consent by the signer, and their exclusive and secure control over the signing device and the keys within. Other factors such as cost and user experience would undoubtedly play a role. Since our focus here is the server-side logic and proof-of-concept for the back-end, we have not considered these issues for the moment.

Other remarks We have assumed that the Service Providers have a constant, resolvable online presence that adequately guarantees a resolvable address for all relevant endpoints, such as eSeal certificate, VC context, refresh, and revocation status endpoints. This seems to us a fair assumption where ASPSP are concerned.

Lastly, in the same way that Financial API are undergoing a standardization process, such as the one being carried out by the FAPI working group [10], VCs would benefit from a reference API without regard of the underlying service infrastructure.

Acknowledgments

The authors would like to thank Istituto Poligrafico e Zecca dello Stato (IPZS) for the collaboration on the development of the authentication solution based on the CIE 3.0 carried out in the context of the joint laboratory DigimatLab between FBK and IPZS.

The research has been partly supported by CherryChain S.r.l. in the context of a research and innovation project funded by Autonomous Province of Trento non-refundable contribution under PAT - APIAE agency resolution n. 333 of 18/12/2019.

This work has been partly developed in the context of the Integrated Framework for Predictive and Collaborative Security of Financial Infrastructures (FINSEC) project, which receives funding from the European Union's Horizon 2020 Research and Innovation Programme under Grant agreement 786727.

References

1. Armando, A., Carbone, R., Compagna, L., Cuéllar, J., Pellegrino, G., Sorniotti, A.: An authentication flaw in browser-based single sign-on protocols: Impact and remediations. Computers & Security **33**, 41 – 58 (2013). https://doi.org/10.1016/j.cose.2012.08.007
2. Banca d'Italia: Disposizioni in materia di adeguata verifica della clientela per il contrasto del riciclaggio e del finanziamento del terrorismo (07 2019), https://www.bancaditalia.it/compiti/vigilanza/normativa/archivio-norme/disposizioni/20190730-dispo/index.html, Content only available in Italian
3. BankID, https://www.bankid.com/en/

4. Berlin Group: NextGenPSD2 Access to Account Interoperability Framework - Implementation Guidelines (07 2019), https://www.berlin-group.org/nextgenpsd2-downloads
5. Bisegna, A., Carbone, R., Martini, I., Odorizzi, V., Pellizzari, G., Ranise, S.: Micro-Id-Gym: Identity management workouts with container-based microservices. International Journal of Information Security and Cybercrime **8**(1), 45–50 (06 2019). https://doi.org/10.19107/IJISC.2019.01.06
6. Deloitte: Value proposition of eIDAS-based eID - banking sector (07 2018), https://ec.europa.eu/cefdigital/wiki/display/EIDCOMMUNITY/Study+on+the+opportunities+and+challenges+of+eID+for+Banking
7. eIDAS eID Technical Subgroup: eIDAS SAML Attribute Profile (07 2014), https://ec.europa.eu/cefdigital/wiki/display/CEFDIGITAL/eIDAS+eID+Profile
8. ETSI: Electronic Signatures and Infrastructures (ESI); Sector Specific Requirements; Qualified Certificate Profiles and TSP Policy Requirements under the payment services Directive (EU) 2015/2366 (11 2019), https://www.etsi.org/standards-search#page=1&search=TS119495
9. European Parliament: Regulation 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. http://data.europa.eu/eli/reg/2014/910/oj
10. Financial-grade api (FAPI) working group, https://openid.net/wg/fapi/
11. Grassi, P.A., Newton, E., Fenton, J.L., Perlner, R., Regenscheid, A., Burr, W., Richer, J., Lefkovitz, N., Danker, J., Choong, Y.Y., Greene, K., Theofanos, M.: Digital Identity Guidelines: Authentication and Lifecycle Management. NIST (06 2017). https://doi.org/10.6028/NIST.SP.800-63b, https://csrc.nist.gov/publications/detail/sp/800-63b/final
12. IPZS: Accesso ai servizi in rete mediante la CIE 3.0 - Manuale operativo per gli erogatori di servizi (04 2020), https://www.cartaidentita.interno.gov.it/identificazione-digitale/entra-con-cie/, content only available in Italian
13. Ministero dell'Interno: Carta dIdentit Elettronica CIE 3.0 - Specifiche Chip (11 2015), https://www.cartaidentita.interno.gov.it/wp-content/uploads/2016/07/cie_3.0_-_specifiche_chip.pdf, content only available in Italian
14. OASIS: SAML V2.0 Tech. Overview. http://www.oasis-open.org/committees/download.php/27819/sstc-saml-tech-overview-2.0-cd-02.pdf (March 2008)
15. W3C: Verifiable Credentials Data Model (11 2019), https://www.w3.org/TR/vc-data-model/

A Listings

Listing 1.1: Example of a Verifiable Credential for the use case described in Section 2.

```
{
  "@context": [
    "https://www.w3.org/2018/credentials/v1",
    "https://<issuer_hostname>/credential/iban"
  ],
  "id": "https://<issuer_hostname>/credential/<
     serialNumber>",
```

```
  "type": ["VerifiableCredential", "ibanCredential"],
  "issuer": "https://<issuer_hostname>/issuer",
  "credentialSubject": {
    "eIDASuniqueIdentifier": <eIDAS unique identifier>,
    "iban": <iban>
  },
  "issuanceDate": <datetime>,
  "expirationDate": <datetime>,
  "credentialStatus": {
    "id": "https://<issuer_hostname>/credential/status
       /<credentialSerialNumber>",
    "type": "OCSP-like"
  },
  "proof": {
"type": <signatureType>,
"created": <datetime>,
        "jws": <jws>,
        "proofPurpose": "assertionMethod",
        "verificationMethod": "https://<issuer_hostname
           >/issuer#nonRepudiationKey"
  }
}
```