H. Roßnagel, C.H. Schunck, S. Mödersheim (Hrsg.): Open Identity Summit 2022,
Lecture Notes in Informatics (LNI), Gesellschaft für Informatik, Bonn 2022 51

Integration of Self-Sovereign Identity into Conventional Software using Established IAM Protocols: A Survey

Michael Kuperberg[1] Robin Klemens [2]

Abstract: Self-Sovereign Identity (SSI) is an approach based on asymmetric cryptography and on decentralized, user-controlled exchange of signed assertions. Most SSI implementations are not based on hierarchic certification schemas, but rather on the peer-to-peer and distributed "web of trust" without root or intermediate CAs. As SSI is a nascent technology, the adoption of vendor-independent SSI standards into existing software landscapes is at an early stage. Conventional enterprise-grade IAM implementations and cloud-based Identity Providers rely on widely established pre-SSI standards, and both will not be replaced by SSI offerings in the next few years. The contribution of this paper is an analysis of patterns and products to bridge unmodified pre-SSI applications and conventional IAM with SSI implementations. Our analysis covers 40+ SSI implementations and major authentication protocols such as OpenID Connect and LDAP.

Keywords: SSI; Self-Sovereign Identity; DID; Decentralized Identifiers; VC; Verifiable Credentials; IAM; Integration; Interoperability; Protocol; OIDC; OpenID Connect; OAuth; SAML; LDAP; X.509 Client Certificates; Kerberos; Active Directory; ADFS

1 Introduction and Problem Statement

Traditional implementations of Identity and Access Management (IAM) in enterprises include products such as Microsoft Active Directory or RedHat KeyCloak, and protocols such as OpenID Connect (OIDC), SAML 2.0, and LDAP. More recently, hosted IAM implementations from cloud-based vendors such as Auth0, Azure or AWS have gained popularity. Still, many companies opt for a hybrid landscape, combining on-premise IAM core deployments with cloud-based applications. Security-wise, PKI (Public Key Infrastructure) standards such as X.509 (incl. certificates for authentication and other tasks) ensure both interoperability and centralized governance, using Certificate Authorities (CAs) and Certificate Revocation Lists (CRLs).

At the same time, a new paradigm has gained momentum *outside* enterprise-internal setups: Self-Sovereign Identity (SSI) [PR21] is a term describing user-centered, user-administered decentralized approach and role model. SSI goes beyond authentication by establishing formats and tools for Verifiable Credentials (VCs). The prevailing implementation of SSI is based on W3C-issued standards for VCs [W3b] and Decentralized Identifiers (DIDs) [W3a].

[1] DB Systel GmbH, Jürgen-Ponto-Platz, 60329 Frankfurt, Germany michael.kuperberg@deutschebahn.com
[2] Institute for Internet Security, Westfälische Hochschule, Germany, and Service-centric Networking, Technische Universität Berlin, Germany, klemens@internet-sicherheit.de

doi:18.18420/OID2022-04

By design, the W3C standards for SSI are substantially different from conventional enterprise-focused IAM and PKI. Consequently, enabling SSI without rewriting existing applications requires additional integration efforts to integrate SSI into the enterprise world and conventional IAM protocols. Furthermore, lack of SSI support at the level of operating systems and web browsers means that additional software has been built for administering DIDs and VCs on devices, resulting in software-based *SSI wallets* such as Lissi-Wallet [Li]. OS-level support or even direct HW support for DIDs and VCs may arrive in the future.

The contribution of this paper is an analysis of solutions which enable the integration of SSI into IAM infrastructures for human users, both for cloud/internet applications and conventional/legacy software. To structure the analysis, we define SSI integration patterns, visualize them and illustrate their impact on the conventional IAM architectures. The analysis of specific products is performed using publicly available information, i.e. the identified software is not subjected to deployment, pilots, assessments, or security analysis.

The paper is structured as follows: Sec. 2 contains the foundations and Sec. 3 presents related work. We define the criteria and the methodology for the evaluation (Sec. 4.1), categorize the 40+ SSI solutions to filter out those which we found to not offer any integration with conventional IAM protocols (Sec. 4.2), and describe the remaining seven products in more detail (Sec. 4.3). Sec. 4.4 presents the comparison results, and Sec. 5 concludes.

2 Foundations

Different architectures and protocols are used for IAM, and new ones are introduced steadily. Yet there is a dominant pattern found in modern web-based applications over the Internet: end users often have the opportunity to login using an account hosted by a separate *Identity Provider* (IdP). The IdP is often owned by a different company, e.g. Google or Facebook, which has many users and mines their data. The resulting "social login" is the public internet equivalent of intra-company Single Sign-On as one IdP can be used across several *Service Providers* (SPs). A the same time, one SP can support a choice of several IdPs.

The SP-IdP pattern is illustrated in Fig. 1 on the left. Authorization is not shown in Fig. 1 because it is implemented very differently depending on the use case: intra-company SSO often includes centralized authorization data, turning an IdP into an IAM system; but certain SPs may keep their authorization data internally as well. In public web applications delivered over the Internet, even when using social login, authorization is mostly kept within SPs. Still, authorization may rely on identity data (such as location, age, gender, etc.) that is stored by the IdP and can be passed to the SP over standardized formats, such as OAuth.

The central precondition for this traditional setup is trust. For web-based applications, trust is established through PKI, specifically through the "trust anchor" set of root CAs. Root CA certificates come pre-bundled with browser downloads or with operating systems. Root CAs issue certificates to intermediate CA, which in turn issue certificates to servers, websites, end users, executable code, etc. CRLs provide additional security.

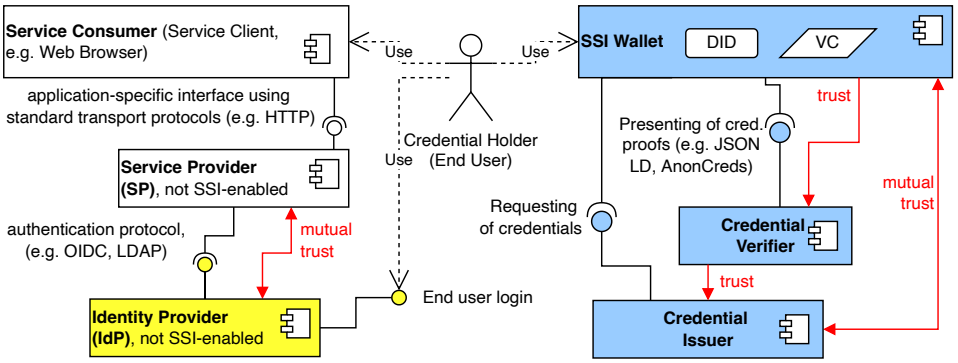

Fig. 1: Non-SSI authentication components (on the left) and SSI components (on the right). Authorization details, PKI infrastructure and SSI Verifiable Data Registry (Identifiers & Schemas) are not shown.

SSI addresses a well-known issue of social logins: while the use of centralized IdPs increases end users' comfort, it also makes end users (and SPs) more dependent on the centralized IdPs (SSI also claims additional benefits, such as machine-readable digitalization of real-life assertions, e.g. possession of driving licenses, etc.). SSI is shown in Fig. 1 on the right. It is also based on asymmetric cryptography but differs by focusing on a *decentralized*, user-controlled exchange of signed assertions. For the explanation of the Issuer, Verifier, Holder and the individual protocols, [PR21] provides a very good in-depth reference while also explaining how SSI fulfills *passwordless login* when implemented natively. Note that the trust relationships for SSI differ from those in the non-SSI case (see red arrows in Fig. 1).

To ensure SSI protocol-level interoperability and to speed up adoption, standards for defining and interchanging DIDs and VCs have been created. In particular, most SSI implementations are not based on hierarchic certification schemas but rather on the peer-to-peer and distributed "web of trust" without root or intermediate CAs. The decentralized approach is also where SSI needs to solve a scaling problem: instead of *limited-scale* "trust anchor" of a few root CAs, all three trust relationships on the right of Fig. 1 must be established for each new wallet holder and/or each new DID - but also for each Credential Verifier and each Credential Issuer. There is no universal solution for this issue, yet this issue is outside our scope.

While the adoption of the existing vendor-independent SSI standards is trying to gain a foothold in the enterprise world (and on the public internet), work on integrating SSI into enterprise environments and landscapes is also far from being completed or standardized. Within enterprise settings (where some services and applications are internal to a company), delegated authentication is often implemented with company-own deployments of identity providers (e.g. Active Directory or RedHat Keycloak). Company-own SSO often extends to both web applications and "traditional" rich clients; it uses such protocols as SAML or Kerberos, but also OAuth and OIDC. The resulting implementation of IAM often exposes

the LDAP protocol interfaces to connect third-party applications to IAM - especially when generic authorization support is needed additionally.

Often, the pre-existing software SP cannot be modified or replaced to support SSI protocols and standards - and even if it could, a co-existence of SSI and non-SSI-based-IAM may require a complex synchronization (e.g. user lockout in IAM must be mirrored in the SSI terms). Also, it is desirable that any changes to the pre-existing IAM solution must be as backward compatible as possible. Consequently, we identified the two following solution patterns that logically focus on IAM functionality:

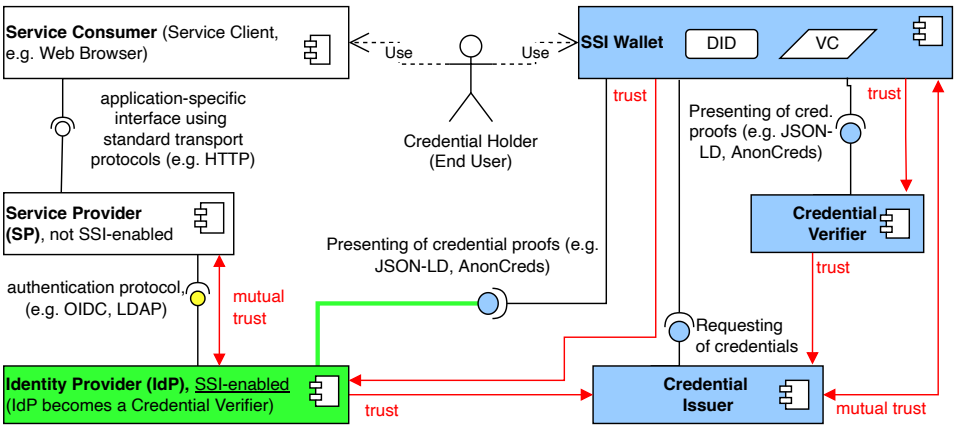

Fig. 2: Pattern A: IdP modified to support SSI protocol(s) for direct interaction with SSI Credential Holders. IdP assumes the role of Credential Verifier; the original Verifier may remain or be removed. Other components remain unchanged. Again, PKI and Verifiable Data Registry are not shown.

- Pattern A: use an SSI-enabled IdP (we illustrate this in Fig. 2) which offers conventional, non-SSI interfaces to the SP but functions as Credential Verifier towards the SSI roles (Holder and Issuer)
- Pattern B: augment the Credential Verifier (we illustrate this in Fig. 3) which provides non-SSI authentication interfaces to the IdP (for "authentication delegation") while leaving the SP *and the IdP* unchanged

Pattern B in Fig. 3 can also be varied into a third pattern, Pattern C, by introducing an *additional* component ("identity broker" or "bridge") between IdP and Verifier. The difference from Pattern C to Fig. 3 (with Pattern B) is that the Verifier remains unchanged while the additional bridge translates between the SSI and non-SSI protocols and data.

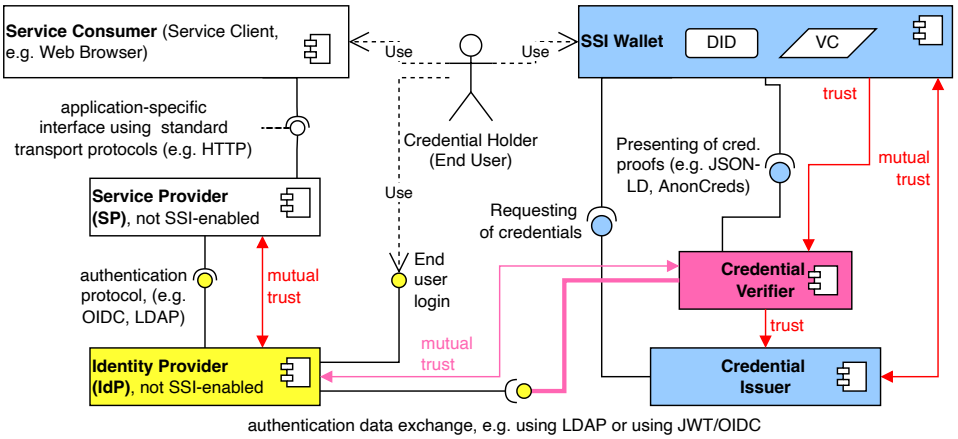

Fig. 3: Pattern B: Credential Verifier modified to support not-SSI protocol(s) for authentication delegation from IdP to Verifier. Pre-existing IdP interfaces are used, and the IdP implementation is not modified. Other components remain unchanged; PKI and Verifiable Data Registry are not shown.

3 Related Work

Several approaches that integrate SSI in/with established non-SSI IAM protocols have already been described. While we analyze individual implementations later in Sec. 4, this section also includes publications describing *comparisons* and surveys of solutions, as well as on theoretical proposals. Among overview-type survey publications such as [ČT21; FCA19; KP; Se21], we haven't found a criteria-driven comparison of SSI products/solutions covering support of traditional authentication/authorization protocols.

Published concepts that have no *publicly available* implementation include [HK20] and [Yi]. Yildiz et al. [Yi] design and implement a prototypical bridge between SSI authentication and SAML-based IdP-to-SP integration. They develop a hybrid solution that switches from username/password to login via VC with minimal authentication flow changes. VaultPoint, the system developed by Hong and Kim [HK20], complies with OAuth2 and combines SSI with smart contracts deployed on Ethereum. The smart contracts allow users to perform authentication and authorization using their own devices. However, the smart contracts store personally identifiable data which cannot be deleted - an approach that is not compliant with the EU GDPR. Thus, we will not consider VaultPoint in the evaluation described below.

Specifications without implementations include [Sa] and [Te]. Sabadello et al. [Sa] describe in their DID Auth document an approach of authentication with a focus on DIDs. In total, the authors present 10 different architectures to complete DID-based authentication by enabling the identity owner to prove control of a DID to a relying party. The OIDC specification "OpenID Connect for Verifiable Presentations" published by Terbu et al. [Te] extends OIDC to support presentation claims over VCs. This allows (1) existing OIDC Relying Parties to

accept VCs as claim sources and (2) new applications built with VCs to use OIDC as an integration layer for credential holders. In addition, the specification enables VC interchange in conjunction with Self-Issued OpenID[3] providers and traditional OICD providers.

Grüner et al. [GMM21] conduct a comparative evaluation of interoperability and portability of schemes for SSI Identity Management Systems (IdMS). As part of their research, the authors analyze the interaction of the user and SP with the IdP. They list OIDC, OAuth2, and SAML 2.0 as traditional IdM compared to DIDAuth and DID as SSI samples. However, their research doesn't evaluate the integration of SSI into traditional IAM protocols.

In constrast to our paper, these eight publications [ČT21; FCA19; GMM21; HK20; KP; Sa; Se21; Te; Yi] do not contain a systematic analysis or comparison of existing offerings.

4 Comparison

4.1 Comparison Methodology Criteria

We evaluate the products based on declared support for six conventional IAM protocols. Comparisons of auth* protocols are frequently made in Internet discussion groups, but we did not identify a *peer-reviewed* publication that would analyze usage frequency of auth* protocols in products, or even compare/rank them. Therefore, we chose the protocols based on our industry experience. The first three (OIDC, SAML 2.0, LDAP) are critically important for a product's relevance and adoption in enterprise environment with preexisting and legacy software, although the specific needs vary in each setting. The remaining three (X.509, Kerberos, AD-native protocols) are less frequent and thus are rather optional. Still, a product supporting one or several of those will be more useful in enterprise settings, in particular where device management is in place (rolling client X.509 for authentication without passwords) or where non-web applications ("fat clients") are in wide use. It should be noted that the protocols we have chosen are neither mutually replaceable nor universal (e.g. LDAP supports the querying of group membership whereas OIDC does not, being restricted to *basic profile infos* [Op]). The main aspects of the individual protocols are as follows:

1. OIDC (an abbreviation for OpenID Connect; it runs on top of OAuth 2.0, resulting in a combination of authentication and authorization), since OIDC is a major integration protocol for web applications, especially on the public internet
2. SAML 2.0 (authentication and authorization), since this is a major SSO protocol for applications in enterprise environments
3. LDAP (incl. LDAPS) (authentication and authorization), since this is the protocol commonly used for legacy centralized IAM in enterprise environments

[3] https://openid.bitbucket.io/connect/openid-connect-self-issued-v2-1_0.html

4. X.509 client certificates (authentication and authorization)
5. Kerberos (only authentication)
6. Active Directory native procotols[4] (authentication and authorization)

As for the comparison itself, we do not perform any tests to verify that a product's advertised features are indeed implemented, and adhere to standards. In other words, we rely on the vendor-provided public information (yet we invested considerable time to clarify incomplete and conflicting statements, and to have the results published). We do not execute black-box or white-box compatibility/functionality tests or even a proper implementation audit.

Note that the interoperation between SSI and established auth* protocols does not cover other essential operational concerns, especially those common to enterprise settings: lifecycle management, compliance, security management, reporting, etc. Also, note that in the comparison below, we have only included SSI solutions which adhere to the W3C standard for DIDs and VCs. There exist further SSI solutions which employ custom protocols (incl. non-disclosed protocols), but we have decided not to include them into the paper's scope because IAM is all about interoperability, exchangeability, and proven standards.

4.2 Filtering out SSI Solutions which do not meet any of the Comparison Criteria

We have studied publicly available information and documentation of 40+ active solutions. Of these, seven solutions claim out-of-the-box support for at lease one of the procotols in Sec. 4.1: (1) SSI Preview in Azure ID, (2) MATTR OIDC Bridge (3) OpenID-SSI_Login, (4) esatus SOWL, (5) Spherity, (6) SSI4A, and (7) VC-OAuthN OIDC.

For the remaining 30+ solutions, we did not identify support for *any* of the conventional IAM protocols that form our six evaluation criteria (cf. Sec. 4.1): (8) Aloaha, (9) Bitnation, (10) Blockchain Helix, (11) cheqd, (12) DigiME, (13) DockIO, (14) Eddits, (15) Element [Mi20b], (16) ESSIF, (17) evan.network, (18) Evernym, (19) Gemalto's SSI effort [Th], (20) "IBM Verify Credentials" (21) ID2020, (22) Identity.com, (23) Idento.one (24) Jolocom, (25) Namecoin, (26) Peaq, (27) selfDID, (28) SelfKey, (29) Seraph ID, (30) Shocard, (31) Sovrin, (32) Serto.id and (33) veramo (two projects created when uPort was split in 2021), (34) SSIBAC, (35) Trinsic (which offers integration with Zapier, but no conventional IAM protocols out-of-the-box), (36) TrustCerts, (37) Veres.one.

Additionally, we found that >10 offerings that used to be active (and have/had websites) appear frozen/abandoned as of Jan. 2022, e.g. Abacus, Block.id, Ethense, FinID, KYC.legal, MHMD, PeerMountain, Persona, Proof/Tierion, Protea, SpidChain, Tenz-ID, etc.

[4] The AD product supports not just LDAP, but also the protocols from `https://docs.microsoft.com/en-us/openspecs/windows_protocols/ms-adod/5ff67bf4-c145-48cb-89cd-4f5482d94664`, such as SAMR/SAMS; the ADFS (Active Directory Federation Services) is an optional add-on that is needed to bring OIDC/OAuth support to AD. Azure Active Directory is a cloud-only offering related to but distinct from AD.

4.3 Compared Products, Implementations, Standards and Initiatives

In this section, we discuss the remaining seven candidate offerings in alphabetical order and summarize our findings in Sec. 4.4 incl. Table 1.

1. *Azure AD verifiable credentials* [Mi20a; Mi21b] is Microsoft's SSI implementation preview that builds on the cloud version of Active Directory (AD); their SSI strategy is found in [Mi21a]. The preview corresponds to Pattern A (cf. Fig. 2). It also partners with multiple identity verification providers to connect the virtual world to the physical world. Concerning adoption, the only mention (as of January 2022) is "the National Health Service (NHS) in the UK". There is no information on when the implementation will leave the preview status and become generally available ("GA").
2. MATTR offers commercial solutions for OIDC-enabled Credential Issuers and Credential Verifiers in JSON-LD format. MATTR OIDC Bridge [MA] is a closed-source extension to MATTR Core with OIDC. The Bridge defines how an OIDC IdP can be extended to support SSI-based authentication leveraging DIDs and VCs and corresponds to our Pattern C (see Sec. 2). The primary function of the Bridge is mapping the presented Credentials claims in JSON-LD format to the OIDC format. At the time of writing this paper, MATTR and the OIDC Bridge only support schema.org for publishing custom data vocabulary; its marked adoption is unclear. However, MATTR stands out by providing pricing information on its public homepage.
3. OpenID-SSI_Login is a prototype described in [Lu20a] by Lux et al., and the source code is open [TU20]. The authors integrate SSI into OIDC: they extend a preexisting IdP and replace the required attributes (within the OICD standard set of claims) with SSI VCs. Thus, OpenID-SSI_Login follows our Pattern A (cf. Fig. 2). [Lu20a] reports that the prototype has been tested with Sovrin and with Hyperledger Indy. No information about the adoption or next releases of OpenID-SSI_Login is available.
4. SOWL [AG] functions as an *Identity Provider* by exposing OIDC, OAuth, SAML 2.0, LDAP, and similar protocol interfaces while maintaining SSI credentials internally. Thus, SOWL follows our Pattern A (cf. Fig. 2) and provides authorization support as well (e.g. over LDAP). SOWL is a closed-source commercial solution and the license fees for the server-side components are fixed by negotiation (the website does not provide any pricing). No information about SOWL's market adoption is available.
5. Spherity Digital Identity Management Toolkit [Sp; St] is a closed-source, commercial SSI implementation targeting IDs for both humans and things (IoT). The Toolkit corresponds to our Patterns A and B (cf. Fig. 2). The toolkit is accompanied by a server-based "Cloud Identity Wallet" (also offered as a SaaS), with an API to connect applications to it and an SDK for integration into mobile apps. In [St], the integration of the Cloud Wallet into IAM landscapes (using LDAP, SAML, OIDC etc.) is announced. The Spherity offerings are marked as General Availability (GA) and pricing is subject to negotiations; no information about market adoption is available.

6. SSI4A [Me] ("SSI for All") is a research project completed in 2019 with no further development since then. The architecture of SSI4A is described in a scientific paper [GMM19] and matches our Pattern A (cf. Fig. 2). The prototype supports uPort and Jolocom (see Sec. 4.2) as SSI solutions. The website says that the users "can obtain attestations about their email address" and names a university portal as the single "integrated application to provide SSI authentication via SSI4A". The source code is not open-source but still publicly available, i.e under a "view only" license.

7. VC-OAuthN OIDC [BC] is another open-source, research-grade project concerned with achieving VC-based authentication using OpenID Connect. As of January 2022, it sees active development and reports tests compatibility with the VON network implementation, using the standardized DIDComm protocol [De] for the messaging between the OpenID Provider and the Identity Holder. The implementation corresponds to our Pattern A (cf. Fig. 2 in Sec. 2). The documentation explains the rationale, architecture and implementation very well. As in [Lu20b], the attributes for the ID token are extracted from VCs provided by the Identity Holder.

4.4 Comparison Results

	Availability; License	OIDC and/or OAuth	SAML 2.0	LDAP	*X.509 client certif.*	*Kerberos*	*AD native*
Azure AD SSI Preview	Preview, cloud-only; commercial	**both**	*no*	*no*	*no*	*no*	**yes**
MATTR OIDC Bridge	GA, cloud-only; commercial	**both**	*no*	*no*	*no*	*no*	*no*
OpenID-SSI_Login	Prototype; open source (ASL 2.0)	**both**	*no*	*no*	*no*	*no*	*no*
SOWL	GA; commercial	**both**	**yes**	**yes**	*no*	*no*	**yes**
Spherity	GA, commercial	**OAuth**	*no*	*no*	*no*	*no*	*no*
SSI4A	Prototype; "read-only" code license	**both**	*no*	*no*	*no*	*no*	*no*
VC-OAuthN OIDC	Prototype; open source (ASL 2.0)	**both**	*no*	*no*	*no*	*no*	*no*

Tab. 1: Native out-of-the-box support of six conventional IAM protocols to connect *applications* to identity providers (see criteria in Sec. 4.1), for the SSI implementations described in Sec. 4.3. Note that Spherity additionally envisions connecting its *Cloud Wallet* to IdPs using OIDC, SAML 2.0 etc.

Table 1 summarizes our findings and we can conclude that *as of January 2022*, support for conventional IAM protocols varies significantly, with OIDC being the most widely supported one, and no Kerberos support. As we did not perform any tests (performance, compatibility, security) and further factors (costs, support, stability/SLAs, etc.) are not considered at this stage, the results do not allow to rank individual offerings or to compare them to each other. It is noteworthy that none of the four commercial offerings is open-source.

5 Conclusions and Future Work

In this paper, we have addressed a key aspect of SSI adoption: which tools and frameworks can support the integration of pre-existing, unmodified applications with SSI concepts and protocols? To start with, we described architectural patterns that can be used for such an integration, by augmenting conventional IAM architectures with additional capabilities and components. Then, we defined a set of protocols (OIDC, SAML 2.0, LDAP, and three others) as criteria for comparing offerings, based on publicly available information.

Of the analyzed 40+ offerings, only seven provide the necessary capabilities by implementing at least one of the necessary protocols. Of these seven, three are research-grade projects, and only one of these three is seeing further development. The remaining four include one preview-status implementation from a major cloud vendor and three GA offerings.

Our research shows that the offerings work in significantly different ways and that there are rather few standardization attempts or best patterns for these aspects, i.e. beyond the SSI protocol level. While we do not recommend a specific product and do not rank the surveyed offerings, we have observed that it is hard to derive information from open documentation and that code examples or integration tutorials are relatively infrequent.

In our future work, we plan to create a reference architecture for integrating pre-SSI architectures with SSI concepts, including authorization aspects. We also plan to perform hands-on tests of the described products to investigate the performance and scalability of hybrid SSI-IAM solutions. Furthermore, we intend to research the intersection of the specifactions of W3C [W3a; W3b] and OIDC [Te] more closely.

Acknowledgements

Andreas Grüner, Axel Küpper, Mirko Mollik, Artur Philipp, and Sebastian Weidenbach supplied very helpful review findings and pointed us to several new initiatives and tools.

References

[AG] esatus AG: esatus SOWL, URL: https://esatus.com/solutions/self-sovereign-identity/sowl/?lang=en, visited on: 01/31/2022.

[BC] BCgov: Verifiable Credential Authentication with OpenID Connect (VC-AuthN OIDC), URL: https://github.com/bcgov/vc-authn-oidc, visited on: 01/31/2022.

[ČT21] Čučko, Š.; Turkanović, M.: Decentralized and Self-Sovereign Identity: Systematic Mapping Study. IEEE Access 9/1, pp. 139009–139027, 2021.

[De] Decentralized Identity Foundation, URL: https://identity.foundation/didcomm-messaging/spec/, visited on: 01/31/2022.

[FCA19] Ferdous, M. S.; Chowdhury, F.; Alassafi, M. O.: In Search of Self-Sovereign Identity Leveraging Blockchain Technology. IEEE Access 7/1, pp. 103059–103079, 2019.

[GMM19] Grüner, A.; Mühle, A.; Meinel, C.: An Integration Architecture to Enable Service Providers for Self-sovereign Identity. In: 2019 IEEE 18th Intl. Symposium on Network Computing and Applications (NCA). Pp. 1–5, Sept. 2019.

[GMM21] Grüner, A.; Mühle, A.; Meinel, C.: Analyzing Interoperability and Portability Concepts for Self-Sovereign Identity./, p. 11, 2021.

[HK20] Hong, S.; Kim, H.: VaultPoint: A Blockchain-Based SSI Model that Complies with OAuth 2.0. Electronics 9/8, 2020, ISSN: 2079-9292, URL: https://www.mdpi.com/2079-9292/9/8/1231.

[KP] Kaneriya, J.; Patel, H.: A Comparative Survey on Blockchain Based Self Sovereign Identity System. In: 2020 3rd International Conference on Intelligent Sustainable Systems (ICISS). IEEE, pp. 1150–1155, URL: https://ieeexplore.ieee.org/document/9315899/.

[Li] LiSSI: The LiSSI wallet - The new solution for Identities. Digital, decentralised and self-sovereign. URL: https://lissi.id/mobile, visited on: 01/31/2022.

[Lu20a] Lux, Z. A.; Thatmann, D.; Zickau, S.; Beierle, F.: Distributed-Ledger-based Authentication with Decentralized Identifiers and Verifiable Credentials. In: BRAINS2020. Pp. 71–78, 2020.

[Lu20b] Lux, Z. A.; Thatmann, D.; Zickau, S.; Beierle, F.: Distributed-Ledger-based Authentication with Decentralized Identifiers and Verifiable Credentials, 2020, arXiv: 2006.04754 [cs.DC].

[MA] MATTR, URL: https://learn.mattr.global/docs/platform/extensions/oidc-bridge/overview, visited on: 01/31/2022.

[Me] Meinel, C.: SSI4A is a gateway to enable the easy use of Self-sovereign Identity (SSI) solutions and a seamless integration into applications, GitHub Repository: https://github.com/agruener2000/ssixa-core, URL: https://ssixa.de, visited on: 01/31/2022.

[Mi20a] Microsoft: Verify once, use everywhere - Join our list and we'll let you know when our Public Preview is ready. 2020, URL: https://didproject.azurewebsites.net, visited on: 12/30/2021.

[Mi20b] Misc.: Element - DID Method implementation using the Sidetree protocol on top of Ethereum and IPFS, https://github.com/decentralized-identity/element, 2020, URL: https://element-did.com, visited on: 01/31/2022.

[Mi21a] Microsoft: Own your digital identity - Discover decentralized identity, a new way to provide ownership of personal data. 2021, URL: https://www.microsoft.com/en-ww/security/business/identity-access-management/decentralized-identity-blockchain, visited on: 01/31/2022.

[Mi21b] Microsoft: Verify once, use everywhere - Use a line of code to verify any data about anyone, while protecting privacy. 2021, URL: https://www.microsoft.com/en-ww/security/business/identity-access-management/verifiable-credentials, visited on: 01/31/2022.

[Op] OpenID Foundation: OpenID Connect Specification, URL: https://openid.net/developers/specs/, visited on: 01/31/2022.

[PR21] Preukschat, A.; Reed, D.: Self-Sovereign Identity. Manning Publications, 2021.

[Sa] Sabadello, M.; Hartog, K. D.; Lundkvist, C.; Franz, C.; Elias, A.; Hughes, A.; Jordan, J.; Zagidulin, D.; Rusu, E.; Powers, A.; Callahan, J.; Andrieu, J., URL: https://github.com/WebOfTrustInfo/rwot6-santabarbara/blob/master/final-documents/did-auth.md, visited on: 01/31/2022.

[Se21] Sedlmeir, J.; Smethurst, R.; Rieger, A.; Fridgen, G.: Digital Identities and Verifiable Credentials. Business and Information Systems Engineering 63/5, pp. 603–613, 2021.

[Sp] Spherity: Spherity, URL: https://spherity.com, visited on: 01/31/2022.

[St] Stöcker, C., URL: https://medium.com/spherity/on-ssi-enabled-idp-solutions-d382abc4b433, visited on: 02/15/2022.

[Te] Terbu, O.; Lodderstedt, T.; Yasuda, K.; Lemmon, A.; Looker, T., URL: https://openid.net/specs/openid-connect-4-verifiable-presentations-1_0-07.html, visited on: 01/31/2022.

[Th] Thales: Self-sovereign identities at work - Digital identity 2.0, URL: https://www.thalesgroup.com/en/markets/digital-identity-and-security/banking-payment/digital-identity, visited on: 12/30/2021.

[TU20] TU Berlin SNET Research Group: OpenID-SSI Login - A bridge between the OpenID-Connect and Self-Sovereign Identity / DIDComm World, 2020, URL: https://github.com/TU-Berlin-SNET/DIMS-openid-ssi_login.

[W3a] W3C: DID (Decentralized Identifier) Data Model and Generic Syntax 1.0, URL: https://github.com/WebOfTrustInfo/rebooting-the-web-of-trust-fall2016/blob/master/draft-documents/DID-Spec-Implementers-Draft-01.pdf, visited on: 01/31/2022.

[W3b] W3C: Verifiable Claims Data Model and Representations, URL: https://www.w3.org/TR/verifiable-claims-data-model/, visited on: 01/31/2022.

[Yi] Yildiz, H.; Ritter, C.; Nguyen, L. T.; Frech, B.; Martinez, M. M.; Kupper, A.: Connecting Self-Sovereign Identity with Federated and User-centric Identities via SAML Integration. In: 2021 IEEE Symposium on Computers and Communications (ISCC). Athens, Greece, pp. 1–7, ISBN: 978-1-66542-744-9.