






Bypassing Current Limitations for Implementing a Credential Delegation for the Industry 4.0

Santiago de Diego¹, Óscar Lage¹, Cristina Regueiro¹, Sergio Anguita¹
and Gabriel Maciá-Fernández²

¹Trustech Department, TECNALIA, Basque Research and Technology Alliance (BRTA), Derio, Spain

²Department of Signal Theory, Telematics and Communications, University of Granada, Granada, Spain

Keywords: IIoT, Identity Management, SSI, Verifiable Credentials.

Abstract: Industry 4.0 is set to modernize industrial processes as we know them today. This modernization goes hand in hand with the digitalization of industry and the need to digitally identify the different devices involved in the manufacturing process. Verifiable credentials and Decentralized Identifiers, which are part of the self-sovereign identity (SSI) concept, allow decentralized identification and characterization of the devices (commonly IIoT devices) that make up Industry 4.0. However, some use cases in the Industry 4.0 cannot be modelled with standard SSI schemes. Despite the fact that delegated credentials have been defined in the W3C standard for verifiable credentials, current technologies present some important limitations that make them non-implementable. This paper analyses these limitations in the context of the problem of building delegated credentials for the Industry 4.0, and proposes an alternative based on an Hyperledger Aries RFC, bypassing these limitations. Finally, some implementation tests have been conducted in order to demonstrate that the Aries RFC does not add extra complexity in terms of performance to the normal SSI flow.

1 INTRODUCTION

The *Fourth Industrial Revolution*, also commonly known as *Industry 4.0*, implies a drastic change in current processes. From the manufacturing side, digitalization plays a crucial role, enabling the so-called *intelligent industry* (Preuveneers & Ilie-Zudor, 2017) because a high level of automation is required to reduce costs to the levels offered by the mass-production paradigm.


Solutions based on the self-sovereign Identity (SSI) framework, where *verifiable credentials* (VCs) (W3C, 2021) and decentralized identifiers (DIDs) (Reed et al., 2021) are utilized in combination with blockchain and are potentially adequate for empowering the Industry 4.0.


The characteristics of a certain manufacturing process, such as the compliance with a standard, are received by different elements manufactured by one


or several factories. In this way, the manufactured element also complies with the standard that the manufacturing process complies with. In other words, the property that the process complied with was received by the elements manufactured in it. It would be much more complex to model this example without using the concept of delegation.


Delegated credentials refer to credentials which allow the user to delegate some of his attributes to another user and is a term recognised by numerous authors (as shown in the State of the Art section), as well as the W3C standard for VCs (W3C, 2021). At the same time, this delegated credential scheme presents challenges in terms of issuing, verifying, and revoking credentials in a chain.


Although the VCs model proposed by the W3C standard defines certain high-level guidelines that can be used for creating delegated credentials, it does not go deep into: *i*) Providing implementation guidelines

^a <https://orcid.org/0000-0002-8823-7509>

^b <https://orcid.org/0000-0003-1168-1932>

^c <https://orcid.org/0000-0002-6031-9449>

^d <https://orcid.org/0000-0001-5517-7645>

^e <https://orcid.org/0000-0001-9256-453X>

for the issuance of delegated credentials. As it will be analysed in Section 4, most of technologies do not currently support this approach; *ii*) Detail how to deal with effective verification of delegated credentials.

This paper improves the current body of knowledge by three contributions. First, it compares and analyses two different approaches for the implementation of delegated credentials; second, it identifies possible incompatibilities when implementing delegated credentials with current SSI technologies; finally, it provides some implementation guidelines to implement the complete flow of a delegated credential.

2 STATE OF THE ART

The SSI paradigm has been broadly studied in the context of Internet of Things (IoT) environments. Some authors (Fedrecheski et al., 2020 ; Mahalle et al., 2020) have deepen on it, presenting own proposals for decentralized identifiers (DIDs) and VCs for the IoT. Most of these works prove that SSI is a better method for identifying IoT devices than other traditional schemes, such as using X.509 certificates. Some authors (Bartolomeu et al., 2019) provide an overview of some SSI concepts and present use cases for the IIoT. Other authors (Niya et al., 2020) have proposed an identification system based on SSI that uses unique information from each IoT device to identify it. In addition, others (Kortesniemi et al., 2019) go further by proposing the use of DIDs as decentralized identifiers for IoT devices and studying the requirements that these devices must meet in order to run an identity management system based on SSI. If these devices are extremely constrained, a proxy-based approach gives support to the solution.

One of the common current issues when implementing SSI solutions is the lack of maturity of SSI technologies (Fedrecheski et al., 2020); the W3C guidelines are clear, but each technology follows its own development roadmap, which may overcomplicate the implementation of SSI-based solutions.

Regarding implementations of SSI, only few SSI-enabling technologies are currently outstanding, including Hyperledger Indy (Windley, 2018) and Hyperledger Aries (Hyperledger Aries, 2020), Veramo (2021)), Veres One (2022), Jolocom (2019), which have different levels of maturity. Indy and Aries have a strong development team supporting them and they are the most mature implementations, followed by Veramo, although they have some

unfinished aspects, as it will be seen later in this paper.

Finally, focusing on the credential delegation concept, some works refer to it. The one in Belenkiy et al. (2009) is likely to be the first work introducing delegated credentials, resulting in a non-feasible implementation mainly due to its complexity, similarly as happened with Chase et al. (2013), as both depend on Groth-Sahai proofs. Camenisch et al. (2017) presented a delegated credentials scheme where the VC is directly passed to a new user that receives delegated permissions to issue new credentials. Then, this issuer can generate copies of this credential, but then signed by himself. Thus, for every credential, he has two instances: the original and the copy. This approach is also suggested by the W3C standard in section C.5. This standard will be discussed in Section 5. Other posterior works (Blömer & Bobolz, 2018; Crites & Lysyanskaya, 2019) study different schemes for delegating credentials, but they are not exempt from limitations; the first scheme does not allow to read the whole delegation history of a credential, while the second is not compatible with adding more attributes to the original credential, which automatically invalidate them for implementing the whole credential delegation framework. Finally, Hyperledger Aries has proposed an RFC with his own method for delegated credentials, termed chained credentials (Hardman & Harchandani, 2021). This method will be also analysed in the next section and we will refer to it as Aries RFC.

3 DELEGATED CREDENTIALS

Initially, let us consider that there is a *certification entity* that issues a credential to another actor, a *manufacturer*. Here, following the SSI approach, the *certification entity* is acting as an *issuer*, and the *manufacturer*, as a *holder*. The *certification entity* would certify that a *manufacturer's* product complies with a certain standard (for example, ISO9001), so it sends to the *manufacturer* a VC that provides evidence that the *manufacturer* does indeed comply with this standard. The *manufacturer* desires to certify some of his *devices*, claiming that they comply with the standard. However, the *manufacturer* only wants to certify the *devices* that belong to a certified product line, specified by the *certification entity*. In other words, not all devices are eligible for receiving the credential. In this case, the *certification entity* must delegate the certification to the *manufacturer*, who will issue a VC to his *devices*. The manufacturer

is in this case acting as a *delegator*, and the device as a *delegate*. It is also desirable that the *certification entity* can express the scope in which this delegation is made. In this context, this credential is termed as delegated credential.

In what follows we analyse W3C standard (W3C, 2021) and the Aries RFC (Hardman & Harchandani, 2021) approaches for building and working with delegated credentials. The W3C standard specifies how delegation must be implemented, while the Aries RFC provides a proposal for delegated credentials, compliant with the W3C.

W3C Standard. The W3C standard proposes expressing the relationship between the delegator and the delegate using a *relationship credential* (RC). Hence, the *verifier* should accept a VP if it includes a delegated credential and the corresponding relationship between the delegate and the delegator. Often, the delegated credential is transferred to the new subject so he can present it together with a copy of this credential, where the *issuer* is the *manufacturer* and the subject is the *device*; and/or a *relationship credential*, which expresses this relationship. The general flow for the W3C proposal has been summed up in Figure 1, where the *manufacturer* is delegating a VC to the *device*. Hence, the *verifier* can ask for a proof directly to the *device*.

Finally, it is worth to mention that W3C proposes the use of their own credentials, known as W3C credentials, which intend to be a standard for VCs. This fact will be important when discussing the implementation issues in Section 4.

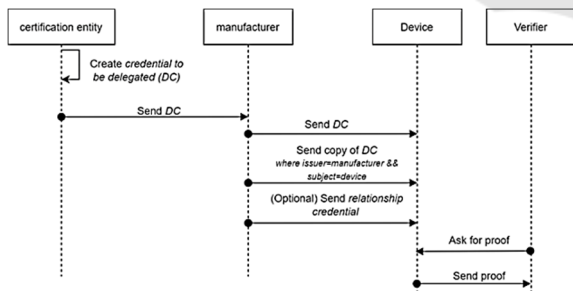


Figure 1: General flow in the W3C standard for delegated credentials.

Aries RFC. Aries proposes, in a Request For Comments (RFC), the creation of a field in the VC containing the VP of the delegated credential, DC. In other words, it proposes including a VP into a VC. Consequently, it proposes creating a chain of trust where the different VPs of the different delegated credentials are attached in a field of the VC, called *provenanceProofs*. The *verifier* then can verify the

VPs included in the VC and therefore accept or reject them if all these proofs are either valid or not. The complete flow of the Aries RFC adapted to the proposed industrial use case is shown in Figure 2.

Comparative Analysis. Regarding privacy, both the W3C and the Aries RFC have their privacy drawbacks. There is a risk of privacy loss when a holder stores another-subject credential, as the W3C proposes. This has also been noted from the research community. For example, some authors (Lim et al., 2021) recognized the possibility of a loss of control when the *subject* delegates a credential to another *holder*; therefore, an authentication mechanism based on public key cryptography is proposed for VCs. However, the verification process becomes complicated because VCs need to be ciphered, and the authors also recognize some drawbacks with this design, e.g., this design is not fully compatible with selective disclosure in VCs. There is also a privacy loss in the Aries RFC because the VP is included in the credential, so the *verifier* has access to all the fields through the VP. However, we consider that it is a better alternative in terms of privacy because the credential is still under the control of his original holder.

The W3C approach might also lead to certain security risks when implemented. In this approach, the *verifier* infers that a group of credentials are related when they are presented together. Therefore, *devices* could steal other credentials from the *manufacturer* and present them with the same *relationship credential*, which would ultimately make the *verifier* accept this credential as valid. In other words, without verifying the original credential against the *manufacturer*, the sole existence of a *relationship credential* does not guarantee full security. Aries RFC does not have this risk because the *holder* only presents one credential containing all the information. However, this design comes with its own drawbacks. By including VPs into VCs, their size is increased, leading into extremely big credentials.

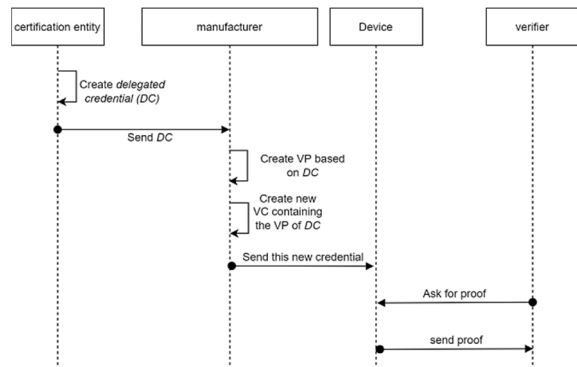


Figure 2: Complete flow for the Aries RFC proposal.

4 ANALYSIS OF CURRENT IMPLEMENTATIONS

From the state of the art, it is conveyed that both Veramo and Hyperledger Indy/Aries are the most advanced technologies for implementing SSI solutions. The implementation of delegated credentials following the W3C standard should be straightforward because all technologies should have support for this type of credentials. However, this is not always true, as some technologies have their own way to implement the standard and some operations, although they are standard-compliant, might not be supported by specific technologies. To validate this assertion, we have carried out some implementation analysis using some technologies, and the conclusions are as follows.

Hyperledger Indy and Hyperledger Aries. A first consideration is related to the Indy wallet. Indy agents do not support storing VCs in the wallet if the subject is different from the *holder* of the credential. This is a technical limitation of Indy-based credentials, which assume that the *holder* and the subject are always the same person. Indy-based credentials do not have a subject field, because they are linked to their *holder*, who is also the subject. Consequently, it is not possible for a delegated *holder* to generate *verifiable presentations (VP)* based on these credentials and, thus, we can deduce that Indy is not compatible with the W3C standard. Regarding Aries wallet, specifically when Indy credentials are used, it simply inherits all the indicated limitations from Indy. However, Aries has recently added support for W3C credentials, which allows to create delegated credentials following the W3C standard.

A second consideration is related to the verification process when the W3C approach is used in Aries. First, it is necessary to implement an additional verification logic to tell the *verifier* that he needs to iterate over the original credentials and check the existence of another credential, which represent the relationship between the *holder* and the subject, before discarding a VP containing a VC whose subject is different from his *holder*. It clearly adds extra complexity when it comes to implementing this protocol. Consequently, it is necessary to make considerable changes within the Indy/Aries code to support this functionality.

Regarding the Aries RFC, Indy currently supports this approach because there is no need to store credentials whose *holder* is different from the subject. Thus, it is compatible with Indy-based credentials. However, Hyperledger Aries must be adapted to

support the Aries RFC because his verification method uses sequence numbers to link the proof request with the rest of the verification process, which does not allow to send the VP, store it and verify it later. This can be indeed useful in order to prevent from replay attacks. By linking the proof request with the rest of the verification process we avoid that a malicious user can reuse the same proof request to obtain the *verifiable presentation* once and again. This mechanism should be implemented by other technologies to prevent this kind of scenarios.

Veramo. Veramo does not present some of the issues that Indy and Aries have. In particular, in Veramo, it is possible that devices store any VC as there are no limitations when it comes to storing credentials whose *subject* is different from the *holder*. This is because Veramo natively works with W3C credentials. This adds more flexibility to Veramo against Indy and Aries. However, regarding the verification process, it is also necessary to implement the same logic in Veramo to properly implement the delegation flow, so some changes need to be made in the source code to do that. Furthermore, Veramo is still in an early stage of development.

Regarding the Aries RFC, it is currently implementable in Veramo, as well as in Hyperledger Indy. Furthermore, Veramo does not implement any sequence numbers between verification steps as Aries does. This can lead to replay attacks as discussed before

Other Technologies. Other technologies, such as those described in the state-of-the-art section, have the same issues. Veres One is still in an early stage of development and the documentation regarding implementation guidelines for VCs is very limited. Jolocom presents a higher maturity level than Veres One, but it still needs to adapt the verification method to support the W3C standard. Regarding the Aries RFC, it is implementable by Jolocom.

Table 1: Mapping between SSI technologies and the W3C (W) and Aries RFC (R).

	Support other-subject VCs	Support verification method	Overall maturity
Indy	R	R	High
Aries	W R		High
Veramo	W R	R	Medium
Veres one			Low
Jolocom	W R	R	Medium

Table 1 maps the current technologies with the W3C standard and the Aries RFC approach regarding delegated credentials. As shown, Aries RFC is supported by most technologies. The overall maturity was determined based on the documentation available for these technologies.

5 PERFORMANCE EVALUATION

Here some results from a performance evaluation of an implementation of delegated credentials based on Hyperledger Indy with the Aries RFC approach are presented. Indy has been chosen instead of Aries so that the described problem of sequence numbers in Aries (see Section 4) is circumvented. Furthermore, Indy is currently more mature than Veramo or Jolocom.

An Indy client that implements the delegation has been implemented in python. A private testing Indy network with four nodes running in a single VM has been deployed as VDR to test the client. The implementation includes these steps: 1) A *certification entity* creates and stores a Credential Schema and an associated Credential Definition; 2) The *certification entity* issues a delegated credential to a *manufacturer* with the Credential Definition; 3) A *verifier* verifies the delegated credential from the *manufacturer*; 4) The *manufacturer* creates and stores a Credential Schema and a Credential Definition for a new VC; 5) The *manufacturer* creates a VP based on his delegated credential and sends the new VC to the *device*, which embeds the VP in an attribute; 6) The *verifier* verifies this new VC; 7) The *verifier* verifies the VP embedded in the VC.

The proposed implementation uses a set of three attributes for the new VC, which are: *i) provenanceProofs*: it stores the VP of the delegated credential; *ii) provenanceSchemas*: it stores the credential schema of the delegated credential; *iii) ProvenanceDefinitions*: it stores the credential definition of the delegated credential.

The *provenanceProofs* attribute solves the problem of storing a VC with a different subject in the wallet and avoid generating a VP based on this VC, which is not possible as discussed in Section 4. The *provenanceSchemas* and *ProvenanceDefinitions* attributes eliminate the necessity of querying the VDR and make the VP self-contained. Additionally, another attribute could be added to control the usage of the delegated credential.

For this purpose, a *nonTransferable* attribute with a Boolean true/false value is used to indicate that this credential cannot be delegated. The *nonTransferable*

property is proposed by the W3C and indicates that a VC must only be encapsulated into a verifiable presentation whose proof was issued by the credential subject. This model can be extended by adding as many attributes as required for more complex scenarios. The labels proposed by Aries RFC are considered to act as separators between different elements and help to retrieve them when required. Thus, we use *[proof]*, *[schema]* and *[definition]* labels to separate them.

Regarding the performance evaluation, the main interest was to analyse if the proposed delegation mechanisms impact in the performance when compared with a scenario without delegation. The latter scenario does not include an embedded VP into the VC and changing the step 7 by a standard verification against the *manufacturer*.

A Dell Latitude 5580 equipped with an Intel® Core™ i7-7600U and 8GM of RAM memory has been used to conduct the tests. For simplicity, all actors are running in the same machine, so network latency does not apply. The experiment has been conducted 400 times and it measures the time spent with and without delegation. The results are shown in Table II. The Proof of Concept (PoC) with delegation corresponds to the seven steps described above, while the PoC without delegation corresponds to the normal Indy flow, which can be found in the Indy documentation.

As seen in Table II, there are not significant differences in terms of performance, so any system supporting the normal Indy flow should support delegation as well with nearly insignificant impact in the performance.

Table 2: Experimental results.

Test	Measure	Value
PoC with delegation	Average	32,73 seconds
	sdv	8,63 seconds
PoC without delegation	Average	32,49 seconds
	sdv	7,53 seconds

6 CONCLUSIONS

This paper has presented the concept of verifiable credential delegation for the Industry 4.0 and has also compared the W3C standard and the Aries RFC proposal for delegated credentials. As it has been analysed, the technologies that allow implementing SSI solutions are not always fully compatible with the W3C standard, which may require minor and, in some of them, major adaptations. The present work has revealed this reality, and some implementation-

related drawbacks have been presented. Finally, a Proof of Concept about delegated credentials has been implemented following the Aries RFC proposal, which avoids some implementation issues for most technologies and presents better characteristics in terms of privacy.

ACKNOWLEDGEMENTS

This work has been supported by Izertis through the SSI4.0 project, which is a collaborative project co-funded by the Department of Economic Development, Sustainability and Environment of the Vice-Ministry of Technology, Innovation and Competitiveness of the Basque Government within the HAZITEK program. (File: ZE-2020/00020).

This work has also been partially supported by Spanish Government-Ministry of Science and Innovation through project SICRAC (PID2020-114495RB-I00).

REFERENCES

- Preuveneers, D., & Ilie-Zudor, E. (2017, April). The intelligent industry of the future: A survey on emerging trends, research challenges and opportunities in Industry 4.0. *Journal of Ambient Intelligence and Smart Environments*, 9(3), 287-298.
- W3C. (2021, November). Verifiable Credentials Data Model 1.1: Expressing Verifiable Information on the Web. <https://www.w3.org/TR/vc-data-model/>. Accessed on: Feb 21, 2022.
- Reed, D., Sporny, M., Longley, D., Allen, C., Grant, R., Sabadello, M., & Holt, J. (2021, August). Decentralized identifiers (DIDs) v1. 0. Draft Community Group Report. [Online]. Available: <https://www.w3.org/TR/did-core/>. Accessed on: Feb 21, 2022.
- Fedrechski, G., Rabaey, J. M., Costa, L. C., Ccori, P. C. C., Pereira, W. T., & Zuffo, M. K. (2020, June). Self-sovereign identity for IoT environments: a perspective. In *2020 Global Internet of Things Summit (GIoTS)* (pp. 1-6). IEEE.
- Mahalle, P. N., Shinde, G., & Shafi, P. M. (2020, February). Rethinking decentralised identifiers and verifiable credentials for the Internet of Things. In *Internet of Things, Smart Computing and Technology: A Roadmap Ahead* (pp. 361-374). Springer, Cham.
- Bartolomeu, P. C., Vieira, E., Hosseini, S. M., & Ferreira, J. (2019, September). Self-sovereign identity: Use-cases, technologies, and challenges for industrial IoT. In *2019 24th IEEE International Conference on Emerging Technologies and Factory Automation (ETFA)* (pp. 1173-1180). IEEE.
- Niya, S. R., Jeffrey, B., & Stiller, B. (2020, November). KYoT: Self-sovereign IoT Identification with a Physically Unclonable Function. In *2020 IEEE 45th Conference on Local Computer Networks (LCN)* (pp. 485-490). IEEE.
- Kortesniemi, Y., Lagutin, D., Elo, T., & Fotiou, N. (2019, March). Improving the privacy of IoT with decentralised identifiers (DIDs). *Journal of Computer Networks and Communications*, 2019.
- Windley, P., & Reed, D. (2018, January). Sovrin: A protocol and token for self-sovereign identity and decentralized trust. Whitepaper, The Sovrin Foundation.
- Hyperledger Aries. (2020). [Online]. Available: <https://github.com/hyperledger/aries>. Accessed on: Feb 21, 2022.
- Veramo – Performant and modular APIs for Verifiable Data and SSI. 2021. [Online]. Available: <https://veramo.io/>. Accessed on: Feb 21, 2022.
- Veres One - A Globally Interoperable Blockchain for Identity. 2022. [Online]. Available: <https://veres.one/>. Accessed on: Feb 21, 2022.
- Jolocom - A Decentralized, Open Source Solution for Digital Identity and Access Management. [Online]. 2019. Available: <https://jolocom.io/wp-content/uploads/2019/12/Jolocom-Whitepaper-v2.1-A-Decentralized-Open-Source-Solution-for-Digital-Identity-and-Access-Management.pdf>. Accessed on: Feb 21, 2022.
- Belenkiy, M., Camenisch, J., Chase, M., Kohlweiss, M., Lysyanskaya, A., & Shacham, H. (2009, August). Randomizable proofs and delegatable anonymous credentials. In *Annual International Cryptology Conference* (pp. 108-125). Springer, Berlin, Heidelberg.
- Chase, M., Kohlweiss, M., Lysyanskaya, A., & Meiklejohn, S. (2013, March). Malleable Signatures: Complex Unary Transformations and Delegatable Anonymous Credentials. *IACR Cryptol. ePrint Arch.*, 2013, 179.
- Camenisch, J., Drijvers, M., & Dubovitskaya, M. (2017, October). Practical UC-secure delegatable credentials with attributes and their application to blockchain. In *Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security* (pp. 683-699).
- Blömer, J., & Bobolz, J. (2018, July). Delegatable attribute-based anonymous credentials from dynamically malleable signatures. In *International Conference on Applied Cryptography and Network Security* (pp. 221-239). Springer, Cham.
- Crites, E. C., & Lysyanskaya, A. (2019, March). Delegatable anonymous credentials from mercurial signatures. In *Cryptographers' Track at the RSA Conference* (pp. 535-555). Springer, Cham.
- Hardman D., Harchandani, L. Aries RFC 0104: Chained Credentials. (2021). [Online]. Available: <https://github.com/hyperledger/aries-rfcs/tree/main/concepts/0104-chained-credentials>. Accessed on: Feb 21, 2022.
- Lim, S., Rhie, M. H., Hwang, D., & Kim, K. H. (2021, January). A Subject-Centric Credential Management Method based on the Verifiable Credentials. In *2021 International Conference on Information Networking (ICOIN)* (pp. 508-510). IEEE.