frontiers in Blockchain

DATA REPORT
published: 30 April 2021
doi: 10.3389/fbloc.2021.622090

Evaluating Trust Assurance in Indy-Based Identity Networks Using Public Ledger Data

Will Abramson[1]*, *Nicky Hickman*[2] *and Nick Spencer*[3]

[1] Blockpass Identity Lab, Edinburgh Napier University, Edinburgh, United Kingdom, [2] Sovrin Foundation, Provo, UT, United States, [3] Incudeas Limited, Thatcham, United Kingdom

Keywords: self-sovereign identity, distributed ledger analytics, verifiable credentials, cryptography, verifier, trust, assurance, risk

1. INTRODUCTION

Self-sovereign identity (SSI) encapsulates a set of technologies, tools, and governance models designed to outline and facilitate the transition to a new paradigm for digital identity systems. One where individuals, organisations, and things are able to actively participate as peers in the digital relationships they establish and maintain over time. The evolving ideology around this movement, initially articulated as 10 principles (Allen, 2016), focuses on empowering the individual, providing them with an independent digital existence that is usable and useful across contexts.

The technical architecture that is emerging defines three distinct transactional roles that entities within an SSI system can engage in; issuer, verifier, and holder. An issuer signs a set of attributes they are attesting to about an entity then presents a data object containing this signature and attributes, a credential, to the entity in the role of holder for this interaction. A holder can then present these attributes along with a cryptographic proof to any number of entities in future interactions. The entity receiving this proof and verifying its integrity is defined as the verifier for that interaction (Sporny et al., 2019a).

To support this architecture a number of open standards are under development. The most mature is the Verifiable Credential Data Model, a W3C recommended standard for the structure of the credential data object that issuers sign (Sporny et al., 2019a). Decentralised Identifiers (DIDs) are another key specification currently going through standardisation in the W3C DID Working Group. This specification defines a new type of identifier designed to facilitate this verifiable, decentralised architecture for digital identity (Reed et al., 2020). DIDs enable entities to provision and manage their own identifiers using a decentralised system and public key cryptography rather than external parties (Allen et al., 2015). These identifiers must be resolvable to a DID Document which contains public keys and authentication mechanisms that support the cryptographic verification of signatures made by the entity in control of the associated private keys.

The W3C DID specification is designed to be technology and protocol agnostic, instead defining a common syntax that can be used to understand all DIDs and a generic set of requirements for create, read, update, and deactivate operations of DID Documents (Reed et al., 2020). Implementers of DID methods select an infrastructure they trust to store these identifiers and their related documents. A distributed ledger, as an append only, immutable, highly available decentralised data storage system is ideal for this infrastructure (Allen et al., 2015; Evans-Greenwood et al., 2016).

This paper focuses on a specific type of distributed ledger designed to support this technical architecture, Hyperledger Indy. The data contained within this ledger are analysed from the perspective of a verifier attempting to assess the risk associated with accepting a credential presentation they have received.

OPEN ACCESS

Edited by:
Michael Shea, Independent Researcher, Litchfield, United States

Reviewed by:
Andreas Freund, Independent Researcher, San Diego, United States
Meghana Kshirsagar, University of Limerick, Ireland

***Correspondence:**
Will Abramson
will.abramson@napier.ac.uk

Specialty section:
This article was submitted to Blockchain for Good, a section of the journal Frontiers in Blockchain

Received: *27 October 2020*
Accepted: *22 March 2021*
Published: *30 April 2021*

Citation:
Abramson W, Hickman N and Spencer N (2021) Evaluating Trust Assurance in Indy-Based Identity Networks Using Public Ledger Data. Front. Blockchain 4:622090. doi: 10.3389/fbloc.2021.622090

2. METHOD

Hyperledger Indy is an open source code base for the instantiation of a ledger to support the creation of public identifiers, DIDs, able to issue and revoke cryptographic credentials using an RSA based scheme first published by Camenisch and Lysyanskaya (2001, 2002). Anyone with read access to the ledger can verify signatures made by issuers on credentials, or their presentations. As Indy has been designed solely for the purpose of identity management and supports anonymous credential cryptography, it stores unique data in contrast to other ledgers that store decentralised identifiers, such as the Bitcoin or Veres One ledgers (Allen et al., 2019; Sporny et al., 2019b). These data are written to the ledger in a number of different transaction types. These are:

- **NYM**—These transactions write a new DID and related DID Document to the ledger.
- **ATTRIB**—Transactions that update existing DID Documents on the ledger, such as rotating keys or changing service endpoints. These must be authored and signed by the DID that identifies the DID Document being updated.
- **SCHEMA**—These transactions define a schema name, version, and list of attribute names for a specific credential. The schema name must be unique on the ledger, but can be altered by writing a schema with the same name and different version number. Versioning a schema must be done by the original author of the schema transaction.
- **CLAIM_DEF**—Often referred to as a credential definition, these transactions write the public key from a generated key pair of an CL-RSA signature for a specific credential schema (Camenisch and Lysyanskaya, 2002). Only DIDs with CLAIM_DEF transactions for specific schema included in the ledger can issue credentials of this schema that are publicly verifiable. Many DIDs can author CLAIM_DEF transactions referencing the same schema.
- **REVOC_REG_DEF**—Transactions that define a revocation registry for a certain credential definition transaction (CL-RSA public key) meaning that credentials signed by this public key can be revoked. Currently, these registries use cryptographic accumulators defined in a 2009 paper by Camenisch et al. (2009).
- **REVOC_REG_ENTRY**—Whenever an issuer issues or revokes a credential, they must author a transaction that updates the revocation registry keeping them up to date so they can be used to construct and verify proofs of non-revocation.

Only NYM and ATTRIB transactions are analogous to other ledgers storing and maintaining DIDs. The reason Indy ledgers include SCHEMA and CLAIM_DEF transactions is likely determined by the need to efficiently support CL-RSA signatures. They have public keys that grow linearly in size with the number of attributes being signed and can take seconds to generate (Camenisch and Lysyanskaya, 2002; Pointcheval and Sanders, 2016). This is too long to be generated at verification time from a single key, hence they are pre-generated by issuers who specify the number of messages to be signed by identifying the schema they intend to issue. This is then stored on the ledger improving verification efficiency. The revocation transactions are similarly unique to Indy ledgers, to our knowledge the only ledger attempting to support anonymous revocation of credentials. These design choices, heavily influenced by the cryptographic primitives the ledger supports, present a richer source of transaction data than other ledgers used to support SSI interactions. As new, more efficient cryptographic protocols, such as BBS+ are supported by Indy, it is expected that the design choices of these ledger will not be so dependent on these protocols (Camenisch et al., 2016).

All transactions include the time they were authored and a unique identifier that can be used to reference and resolve data from within them. Transactions must be signed by the public key associated with a DID already stored on the ledger before it is accepted by the nodes maintaining the ledger state. This leads to a hierarchical structure whereby all DIDs must first be authored to the ledger in a nym transaction signed by the key of another DID before they can themselves write transactions to the ledger. Any Indy-based ledger is initiated with a number of genesis nym transactions and all other nym transactions can be traced to a nym transaction signed by one of these DIDs. This structure of signed transactions allows any entity to verify the validity of the ledger state by starting from these genesis transactions. It also ensures rules around which DID has the authority to update a DID Document, schema, or revocation registry can be cryptographically enforced.

The dataset under analysis in this paper are the transactions from a specific instantiation of an Indy based distributed ledger, the Sovrin MainNet. A ledger that has been running since July 2017 that supports some of the most mature deployments SSI systems today. The ledger includes 448 nym transactions, including 16 genesis nyms representing the Sovrin board of trustees, 88 schema, and 356 credential definitions. While other Indy ledgers include far more transactions, such as the Sovrin StagingNet with almost 20,000 nym transactions, the Sovrin MainNet is for production deployments of SSI so provides a more realistic dataset. Despite this focus on the MainNet, the analysis should be at least partially applicable to any Indy-based distributed ledger.

A major difference between the Sovrin MainNet and other Indy networks is that it is a public-permissioned network governed by the Sovrin Governance Framework that defines the roles and responsibilities of different actors within the network (Sovrin Governance, 2019a). A permissioned network adds additional constraints around who can write to the ledger. In the Sovrin MainNet only DIDs with the role of transaction endorser are able to write nym transactions to the ledger and all subsequent transactions these DIDs author must be additionally signed by a transaction endorser (Sovrin Governance, 2019c). This presents interesting opportunities for analysis as we discuss later in the paper.

The nodes within the Sovrin MainNet are run by Sovrin Stewards, organisations that volunteer time and resources to maintain the network. The network is administered and managed by the Sovrin Foundation which also acts as a Governance Authority (Sovrin Governance, 2019a). Stewards are selected

by the Governance Authority to ensure maximal distribution of hardware, domain, and geographic location limiting the threat vector of malicious takeover and promoting resilience. All stewards agree to the requirements specified by the Governance Framework and sign the Sovrin Stewards Agreement (Sovrin Governance, 2019a,b). Nodes accepted into the network then engage in a consensus protocol named plenum based on redundant byzantine fault tolerance (Aublin et al., 2013). As such, assuming the Sovrin Foundation and a subset of the stewards can be trusted then the transactions stored within the ledger can be trusted with a high degree of confidence.

The data discussed within this paper can be accessed through the public hyperledger indy transaction explorer, IndyScan. For more detailed analysis, it is also possible to clone the github repository for this explorer and visualise the data from the ledger through a Kibana dashboard or similar. Alternatively, the ledger data can be fetched using indy-vdr a hyperledger repository designed for querying indy nodes. This paper uses visualisations of a subset of MainNet transactions retrieved using the IndyScan API.

3. ANALYSIS

Analysis of the data held within a Hyperledger Indy network may be useful for answering questions from many different perspectives within an SSI system. This paper focuses on one in detail, that of a verifier attempting to determine whether to accept a proof of a set of attributes presented by a credential holder. While this decision will be tied to the semantic context of the interaction and is largely subjective for each verifier, we focus our analysis specifically on the syntax, the information contained within the ledger that might influence the decision of a verifier. Either alerting them to increased risk, or giving them a greater degree of assurance.

The presentation of indy-backed credentials is specified by Aries-rfc-0037 (Khateev, 2019), a protocol involving two entities, a holder and a verifier, that have previously exchanged peer DIDs to establish a DIDComm channel across which encrypted, digitally signed messages can be exchanged, authenticated, and decrypted. The holder then constructs a proof object from a set of credentials that have previously been issued to them and sends this to the verifier. From this proof, the verifier is able to learn:

- The attribute values presented
- The identifiers of the scheme the attributes were issued in
- The identifiers of a set of claim definitions
- The mathematical proof of the integrity of the attributes
- The mathematical proof of a common master secret attribute known to the holder and signed by the issuer of each credential involved in the presentation
- The identifiers for the revocation registries of credentials if applicable

The verifier can then query the ledger for the CLAIM_DEF transactions to return the public keys of the issuers of each of the credentials used to construct the proof. Using these keys the proof object can be mathematically verified such that the verifier can have high confidence that the attributes presented were issued to the same master secret, the holder knows this secret and the attributes presented have not been tampered with since issuance. Additionally, resolving the REVOC_REG_DEF transactions allows for verification of any proof of non-revocation, if this has been included in the presentation. However, in addition to the fidelity of the information contained within the presentation, a verifier must assess its provenance (Windely, 2020).

This paper suggests Indy transaction data can provide insights into the question of provenance by using the SCHEMA and CLAIM_DEF transaction identifiers as a starting point for inquiry. By querying the ledger dataset for these transactions, the verifier learns the DIDs of the transaction author and transaction endorser for both transactions. Depending on the context, different comparisons may be appropriate here. A verifier may expect both of these transactions to have been endorsed by the same DID. In the future, this may present a mechanism to associate a presentation with a specific governance domain that the credentials were issued under, where the endorser represents a governance authority. In contrast, when comparing the DID that authored the SCHEMA with that of the CLAIM_DEF, a difference here might give the verifier greater assurance.

Another potentially useful insight can be gained from the ledger by querying all CLAIM_DEF transactions that reference the schema used within the presentation. See the dotted lines between blue nodes (SCHEMA) and green nodes (CLAIM_DEF) in **Figures 1**, **2**. Through this, the verifier learns how many distinct issuers are able to issue this credential, giving some indication of its value and adoption. This analysis can be extended further by including the transaction endorsers of these CLAIM_DEF transactions and, further still, to include the endorser of the NYM transactions for the DIDs that authored these CLAIM_DEFs. A visualisation of this analysis can be seen in **Figure 2**.

This approach effectively graphs the roots of trust associated with a particular credential schema. In this instance, a single endorser used for all transactions might indicate a strong governance domain, particularly where there are many issuers involved. The analysis of these patterns can be derived from the SCHEMA transaction identifier, information that is included in a presentation request so available to all verifiers. Additionally, by placing the CLAIM_DEF and NYM transactions of the issuer within this pattern it may be possible to spot anomalies alerting them of potentially untrustworthy issuers. For example, if these transactions had been endorsed by a different DID in a schema pattern that has a common endorser for all other transactions. Such patterns can clearly be seen within the Sovrin MainNet, as the visualisations in **Figures 1**, **2** show.

Querying the ledger for information about a DID could be worthwhile for certain verifiers as it would enable them to see all the transactions they have authored over time. The importance of the author of the NYM transaction that initially wrote this DID to the ledger has already been emphasised, however, other information may be equally useful. For example, how long ago the NYM transaction was authored, how many CLAIM_DEF transactions they have written to the ledger, and which credential schema are they for.

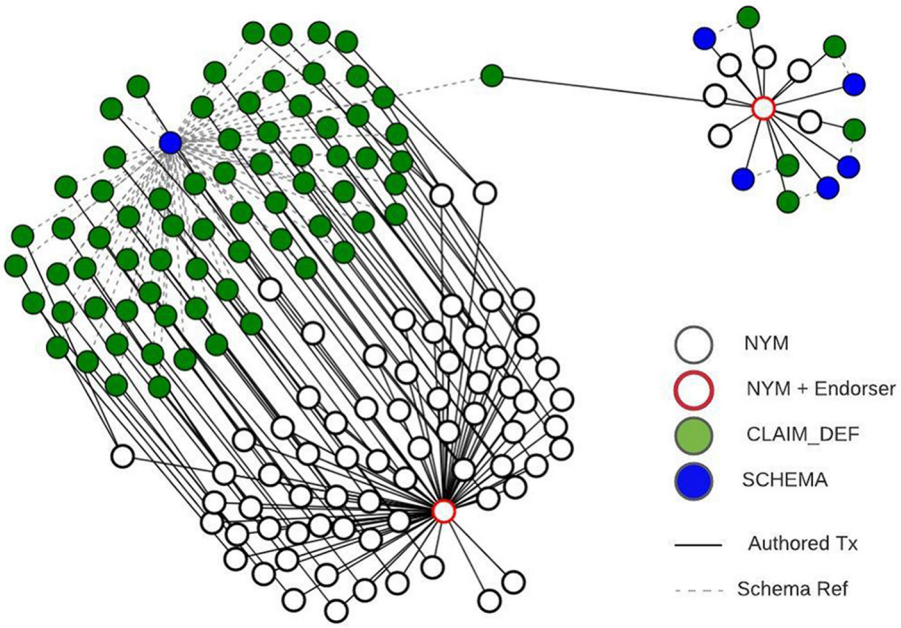

FIGURE 1 | Visualisation of authored transactions linked to a single schema identifier.

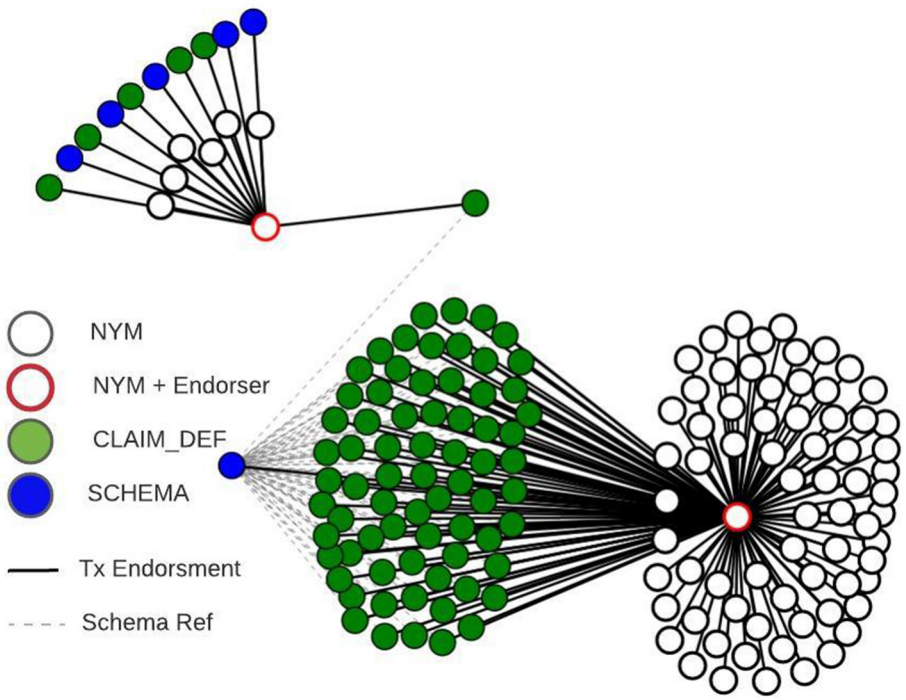

FIGURE 2 | Visualisation of transaction endorsement for the same transactions shown in **Figure 1**.

The analysis presented has focused only on the ledger data, following a logical pathway of inquiry a verifier might take when presented with a proof object from an entity containing SCHEMA and CLAIM_DEF transaction identifiers. It has been described to illustrate what it is possible to learn from this data independently of any contextual information that can be inferred from the interaction or provided by the verifying entity itself. This additional information may determine which questions are

appropriate to ask from the data, as well as the acceptable answers a verifier expects. An example of this might be the expectation that issuers NYM and CLAIM_DEF transactions were endorsed by a specific DID that is meaningful to the verifier.

4. CONCLUSION

This paper takes an indepth look at the data available within Hyperledger Indy-based ledgers, focusing particularly on the Sovrin MainNet, an established public ledger designed for production use cases. This specific instantiation has well-defined governance processes and legally binding agreements for all actors within the network. Assuming trust is placed in these processes then the information within the ledger can be trusted to a high degree of assurance. In the future, it is expected that many more public networks based on Hyperledger Indy will emerge for production use cases, as this happens the ability to assess the trust placed in the specific ledger itself will become increasingly important. This work is already underway within the Sovrin community to define a set of common metrics with which to evaluate different Indy nodes, ledgers, and networks (Foundation, 2020; Indy, 2020).

For now though, it is important to recognise that the ledger within an SSI network is designed to be a highly assured source of information. Wherever there is data, there are insights that can be drawn from this data. This paper puts forward an initial attempt to describe exactly what these insights might be and how they could be useful from the perspective of a verifier. Within SSI, there are many perspectives that could adapt the approaches described within this paper to answer their own questions. Implications of this research could be built into the governance framework's assurance policies as well as verifiers' business logic and user experience design. Equally, this suggests that information from a public Indy ledger has potential privacy and security implications for issuers. Further research is required here, but it may be that for certain use cases and industries, this is unacceptable.

We emphasise that this report is focused primarily on the structure of the transaction data found within Indy ledgers and the potential patterns that might emerge when these transactions and their relationships are graphed. While the use case visualised in **Figures 1**, **2** are of real transaction data on the Sovrin MainNet from an advanced pilot within healthcare known to the authors, it has been presented to illustrate the kinds of relationships and patterns we think are useful to pay attention to. It is our hope that this work stimulates further research into the patterns found across a statistically meaningful sample of SSI applications, so that reliable conclusions can be drawn.

In addition to this, there are many other DID methods that resolve identifiers against other distributed ledgers, such as Bitcoin, Ethereum, and Veres One. These are all permissionless ledgers that support decentralised identity systems without storing schema or credential definitions on the ledgers, a quirk of Indy-based ledgers due to the anonymous credential cryptography they support. This means that DIDs will not be so directly correlated with the schema they can issue, or schema with the DIDs that can issue them. Furthermore, since anyone can write a DID to permissionless ledgers, different mechanisms will need to be implemented to determine a DIDs provenance. Finally, credential systems using non-Indy ledgers often require holders to record DIDs on the ledger in order to be able to authenticate as the credential subject to a verifier. The advantages and disadvantages of these differences and their implications for potential ledger analysis deserve further attention.

DATA AVAILABILITY STATEMENT

Publicly available datasets were analyzed in this study. This data can be found at: https://indyscan.io/home/SOVRIN_MAINNET, https://github.com/wip-abramson/sovrin-network-vis, https://github.com/wip-abramson/sovrin-network-vis/blob/master/src/graphdata.json.

AUTHOR CONTRIBUTIONS

WA, NH, and NS have been collaborating on broader research into metrics for SSI systems for over 6 months. This work informed all ideas presented in this paper. WA analysed the ledger data, created the visualisation, and wrote up the first draft. NH and NS provided the feedback and review. All authors contributed to the article and approved the submitted version.

ACKNOWLEDGMENTS

The authors would like to acknowledge Patrik Stas, the creator of the IndyScan transaction explorer who helped us access and interpret the data analysed in this paper.

REFERENCES

Allen, C. (2016). *The Path to Self-Sovereign Identity*. Life with Alacrity. Available online at: http://www.lifewithalacrity.com/2016/04/the-path-to-self-sovereign-identity.html

Allen, C., Brock, A., Buterin, V., Callas, J., Dorje, D., Lundkvist, C., et al. (2015). *Decentralized Public Key Infrastructure*. Group Report. Rebooting the Web of Trust (RWoT). Available online at: https://danubetech.com/download/dpki.pdf.

Allen, C., Duffy, Kim, H., Grant, R., and Pape, D. (2019). *BTCR DID Method*. Technical report, World Wide Web Consortium.

Aublin, P. L., Mokhtar, S. B., and Quéma, V. (2013). "RBFT: redundant byzantine fault tolerance," in *2013 IEEE 33rd International Conference on Distributed Computing Systems* (IEEE), 297–306. doi: 10.1109/ICDCS.2013.53

Camenisch, J., Drijvers, M., and Lehmann, A. (2016). "Anonymous attestation using the strong Diffie Hellman assumption revisited," in *International Conference on Trust and Trustworthy Computing* (Springer), 1–20. doi: 10.1007/978-3-319-45572-3_1

Camenisch, J., Kohlweiss, M., and Soriente, C. (2009). "An accumulator based on bilinear maps and efficient revocation for anonymous credentials," in *International Workshop on Public Key Cryptography* (Springer), 481–500. doi: 10.1007/978-3-642-00468-1_27

Camenisch, J., and Lysyanskaya, A. (2001). "An efficient system for non-transferable anonymous credentials with optional anonymity revocation," in *International Conference on the Theory and Applications of Cryptographic Techniques* (Springer), 93–118. doi: 10.1007/3-540-44987-6_7

Camenisch, J., and Lysyanskaya, A. (2002). "A signature scheme with efficient protocols," in *International Conference on Security in Communication Networks* (Springer), 268–289. doi: 10.1007/3-540-36413-7_20

Evans-Greenwood, P., Hillard, R., Harper, I., and Williams, P. (2016). *Bitcoin, Blockchain & Distributed Ledgers: Caught Between Promise and Reality*. Deloitte.

Foundation, S. (2020). *SSI Metrics Dashboard*. Available online at: https://github.com/hyperledger/indy-node-monitor

Indy, H. (2020). *Indy Node Monitor*. Available online at: https://github.com/hyperledger/indy-node-monitor

Khateev, N. (2019). *Aries RFC 0037: Present Proof Protocol 1.0*. Technical report.

Pointcheval, D., and Sanders, O. (2016). "Short randomizable signatures," in *Cryptographers' Track at the RSA Conference* (Springer), 111–126. doi: 10.1007/978-3-319-29485-8_7

Reed, D., Sporny, M., Longely, D., Allen, C., Sabadello, M., and Grant, R. (2020). *Decentralized Identifiers (DIDs) v1.0*. Technical Report. World Wide Web Consortium.

Sovrin Governance, F. W. G. (2019a). *Sovrin Governance Framework v2*. Provo, UT: The Sovrin Foundation.

Sovrin Governance, F. W. G. (2019b). *Sovrin Stewards Agreement*. Provo, UT: The Sovrin Foundation.

Sovrin Governance, F. W. G. (2019c). *Sovrin Transaction Endorser Agreement*. Provo, UT: The Sovrin Foundation.

Sporny, M., Longely, D., and Chadwick, D. (2019a). *Verifiable Credentials Data Model 1.0*. Technical report, World Wide Web Consortium.

Sporny, M., Longely, D., and Christopher, W. (2019b). *Veres One DID Method 1.0*. Technical report, World Wide Web Consortium.

Windely, P. (2020). *Origins and Principles of SSI*. Breaking Silos.

Conflict of Interest: NS was employed by company Incudeas Limited.

The remaining authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.