Check for updates

RESEARCH ARTICLE

REVISED A blockchain-governed SSI trust architecture for EDC-based federated data spaces

[version 2; peer review: 1 approved]

Lidia Alaejos Herrero[1], María Ruiz Molina[2], Diego Molinero Moreno[1], Javier Prieto Tejedor[1]

[1]Universidad de Salamanca, Salamanca, Castile and León, Spain
[2]DigitelTS, MadisonMK, Valladolid, Castile and León, Spain

v2 **First published:** 22 Jul 2026, **6**:253
https://doi.org/10.12688/openreseurope.24229.1
Latest published: 28 Jul 2026, **6**:253
https://doi.org/10.12688/openreseurope.24229.2

Abstract

The usage of blockchain technologies incorporates a trust layer due to its immutability and transparency for all participants. The implementation of this technology in order to validate and authenticate the identity of different entities in a Data Spaces ecosystem, allows to all the contributors to validate which of them have certain sets of traits suitable for their data sharing, interests, and moreover, role managing.

This way, connectors can automatically identify with who they are interacting and choose which data to share with them based on these characteristics. Some useful examples are the detection of certain issued verifiable credentials that prove their belonging to a specific sector, accreditations ... that automatically provide them with access to specific datasets.

In order to achieve this, the presented Trust Layer uses a governance infrastructure with blockchain, an identity layer to store private keys and connect the blockchain address to a decentralized identifier, and a credential manager to issue and verify the Verifiable Credentials. Added to this, an integration layer is designed and implemented, which connects these components to the Data Space Connector. For this specific technical implementation, the design has been built to be operable with Eclipse Dataspace Components (EDC), given its Verifiable Credentials interoperability and its wide adoption among Data Spaces as their preferred connector.

A qualitative discussion of the architecture's strengths and challenges is provided, covering aspects such as decentralization, regulatory

Open Peer Review

Approval Status ✓

	1
version 2 (revision) 28 Jul 2026	✓ view
version 1 22 Jul 2026	

1. **Miguel Pincheira**, Fondazione Bruno Kessler, Trento, Italy

Any reports and responses or comments on the article can be found at the end of the article.

alignment, and adoption barriers, together with a comparison against alternative identity and trust models for federated data spaces.

Keywords
Self-Sovereign Identity, Blockchain, Federated Data Spaces, Eclipse Dataspace Components, Verifiable Credentials, Trust Architecture



This article is included in the Horizon Europe gateway.

Corresponding authors: Lidia Alaejos Herrero (lidiaah@usal.es), María Ruiz Molina (maria.ruizm@madisonmk.com), Diego Molinero Moreno (diegomm2001@usal.es), Javier Prieto Tejedor (javierp@usal.es)

Author roles: Alaejos Herrero L: Conceptualization, Data Curation, Formal Analysis, Funding Acquisition, Investigation, Methodology, Resources, Software, Supervision, Writing – Original Draft Preparation, Writing – Review & Editing; **Ruiz Molina M**: Conceptualization, Funding Acquisition, Project Administration, Software, Supervision, Validation, Writing – Original Draft Preparation, Writing – Review & Editing; **Molinero Moreno D**: Investigation, Methodology, Resources, Software, Visualization, Writing – Original Draft Preparation; **Prieto Tejedor J**: Conceptualization, Funding Acquisition, Investigation, Project Administration, Resources, Supervision, Validation, Writing – Original Draft Preparation, Writing – Review & Editing

Competing interests: No competing interests were disclosed.

Grant information: This project has received funding from the European Union's Horizon Europe research and innovation programme under grant agreement No 101135927 (A catalyst for EuropeaN ClOUd Services in the era of data spaces, high-performance and edge computing (NOUS)).
The funders had no role in study design, data collection and analysis, decision to publish, or preparation of the manuscript.

Copyright: © 2026 Alaejos Herrero L *et al*. This is an open access article distributed under the terms of the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

How to cite this article: Alaejos Herrero L, Ruiz Molina M, Molinero Moreno D and Prieto Tejedor J. **A blockchain-governed SSI trust architecture for EDC-based federated data spaces [version 2; peer review: 1 approved]** Open Research Europe 2026, **6**:253 https://doi.org/10.12688/openreseurope.24229.2

First published: 22 Jul 2026, **6**:253 https://doi.org/10.12688/openreseurope.24229.1

REVISED **Amendments from Version 1**

Added in acknowledgements: "This research has been supported by the project "A catalyst for EuropeaN ClOUd Services in the era of data spaces, high-performance and edge computing (NOUS)", Grant Agreement Number 101135927. Funded by the European Union, views and opinions expressed are, however, those of the authors only and do not necessarily reflect those of the European Union. Neither the European Union nor the granting authority can be held responsible for them."

Any further responses from the reviewers can be found at the end of the article

Introduction

The Dataspaces ecosystem has been evolving more and more during the last years, creating several standards, sometimes under a common framework, but often basing their designs on different technology stacks, thus leading to a lack of interoperability between different connectors. Moreover, necessary complements to Dataspaces, such as identity management solutions, are in an even earlier stage.

Among the diverse Dataspace solutions we can find the usage of federated dataspaces, a way to control who can access them by fulfilling a series of requirements before joining the ecosystem. Some of these are Gaia-X, IDSA (International Data Spaces Association) and Catena-X.

The Gaia-X[13] initiative presents a framework that defines the rules for control and governance; and a common set of guidelines and rules to apply to the dataspaces technologies and infrastructure in Europe. It aims to create an environment of trust where participants can share data in an open, transparent and secure way. For instance, Catena-X[14] is a dataspace part of the Gaia-X ecosystem, specifically focused on the automotive industry.

IDSA[16] is an association with the goal of defining an international standard for dataspaces and technologies related to them. Its architecture has been developed by Fraunhofer-Gesellschaft,[17] a Germany-based organization focused on application-oriented research.

Moreover, some institutions, such as the International Data Spaces Association[1,12] or the Data Spaces Support Centre[19] in the European Union, were created to provide a centralized common point of trust to all its participants.

Nonetheless, these federated solutions and centralized institutions create ecosystems where participants must meet a set of requirements, verification process and approval by a centralized entity, before they can join the ecosystems. These decisions determine who can participate in the ecosystem, oftentimes implying geographical restrictions or the belonging to certain trusted groups that some actors might not be able of ever fulfill, remaining outside of the ecosystem. Therefore, this design does not allow everyone to participate equally and leaves all power of decision to the central authority in charge of the federated ecosystem.

Alongside this need to authenticate users in a dataspace ecosystem, there is a necessity to manage their roles once inside of it. These can range from administrative roles to specific attribute-based roles, such as field-belonging, certificates, memberships, etc.

Some other potential solution is the usage of Public Key Infrastructure to generate X.509 certificates.[33] On this scenario, participants would identify themselves through their digital signature using keys issued by an entity recognized by the system. This system by itself is not enough, since all identity attributes' verification rely on the issuer of the keys. This would require an update on the associated certificate information every single time new information about the participant is added, relying fully again on the issuer.

In order to exchange authentication tokens, platforms might use OAuth2 to be authenticated within the ecosystem and OpenID Connect (OIDC) to receive identity tokens. This way users maintain a higher level of control on their shared attributes, but the issuance of these tokens still rely on the party with which the actors must interact in order to be properly authenticated.

The introduction of blockchain usage to reflect the issuing of new attributes on the ledger, actors can have their new credentials issued and linked to them in a transparent and immutable way while not needing to recreate their keys every single time. The information is just managed on-chain and any other participant can verify who issued what about each actor, their level of trust, and all independently of other issued traits.

Because of the decentralized and peer-to-peer (or connector to connector) nature of dataspaces, a Self-Sovereign Identity (SSI) model, that implements both an authorization control and role management, paired with the usage of blockchain technologies, solves several of the issues caused by centralization of authentication.

Because of it, a more decentralized identification and authentication processes are required in order to make the authentication and integration of new actors to dataspaces easier.

This paper presents a blockchain and SSI based identity architecture for Dataspaces. The identity is represented through the Hyperledger Indy decentralized identifiers in the form of did:indy:besu, and a proxy is usage between the connector's core and the blockchain in order to communicate one with the other in a faster manner. The implementation is done specifically with Eclipse Dataspace Components (EDC) (Section 6) as the technology for the connectors in the dataspace.

This paper is organized as follows: first, an introduction (Section 1) providing context; second, a review of the Background and Related Work (Section 2) establishing the current state of the art; subsequently, the Problem Statement and Research Questions (Section 3) are defined to guide the inquiry. Following this, the Blockchain-Governed SSI Trust Model (Section 4) is introduced, followed by a detailed description of the proposed Architecture (Section 5).

Furthermore, the Integration with EDC (Section 6) is detailed to demonstrate technical interoperability; next, the Experimental Methodology (Section sec:methodology) and the corresponding Experimental Results (Section 8) are presented and analyzed. Afterward, a dedicated Security, Governance, and Trust Analysis (Section 9) evaluates the robustness of the system; subsequently, a Discussion (Section 10) and an assessment of the Threats to Validity (Section 11) are provided. Finally, the paper closes with a Conclusion and Future Work (Section 12) to summarize the findings and suggest next research directions.

Background and related work

To contextualize the development of the architecture presented in this paper, we present here the State of the Art of identity and authentication methods in Dataspaces. For it, we must introduce these both concepts initially separated, and later on, how they converge.

Federated data spaces and data sovereignty

Federated Data Spaces create a trust framework so that all participants can share information with each other directly, connecting with the interested parties to initiate a negotiation on the requested data. This allows actors to know all other participants and their offered data in the catalogue of the dataspaces, but to start negotiations only with those in which they have interest on exchanging information.

On this system, actors can choose which information to expose and with what type of other actors. It allows sharing data in a way they can control with whom, when, and what types of information. This architecture matches the way Data Spaces intend to work: giving full autonomy to the participants but at the cost of bigger technological demands on how to communicate and exchange information between each other.

Federated Data Spaces allow to define rules on how the participants interact, data infrastructure and governance,[34] while allowing them to have full autonomy on every operation's decision, and to initiate communication with only the actors they want to.

In the European Union, Federated Dataspaces is the go to architecture for sharing data.[35] Specifically, Gaia-X[13] is the chosen Federated Data Space Architecture, with applications on already ongoing data spaces, such as the Mobility Data Space[18] which is implemented by following the provisions of the International Data Spaces (IDS)[16] Reference Architecture Model (RAM).[20]

Trust and identity management in data spaces

In order to be authenticated in a federated data spaces, participants must comply with a certain set of rules established for the ecosystem itself. These are given to them the moment they join, and used to access the data space from then on. Nonetheless, this access is not enough in order to be authorized to use or visualize certain data types in a catalogue.

For the authorization to access certain information or to take certain actions, the actor must have a specific set of attributes. These can vary from roles within the data space organization to roles regarding specific data types, organization attributes, etc. This set of attributes define the data they can visualize in the catalogue and, therefore, request.

This architecture allows to create subsets of participants based on the type of information they specialize on and their interests. For instance, healthcare entities can share health-related data without having to expose their datasets to non-healthcare participants, while academic entities can expose their education credentials between each other, but only medicine students and professors can request healthcare datasets in the catalogue that are shared by any other health-related entity for academic or research purposes.

Finally, trust is based on who defined those attributes for each participant. Every other actor can verify the origin of the information about another one, allowing them to configure which other participants they trust based on their very own trusted list of issuers. This way, every participant can decide to be more or less flexible in what regards origin of attributes. Some might prefer to trust more entities by having a lower level of trust acceptance, and others might prefer to constrain their interactions with those that have their attributes issued by a highly trusted and official issuer.

Thanks to this, participants can block negotiations requested from entities with attributes issued by entities outside of their trusted list, allowing them to keep full control on their data.

Self-Sovereign identity, DIDs and verifiable credentials

Self-Sovereign Identity introduces a new model where users keep full control of their identity and data associated to it. This puts the main focus on privacy and allows users to know with who are they sharing which data, being selective about the information they disclose to each entity they interact with. To achieve this, models use Verifiable Presentations[7] to choose the traits they want to disclose, and zero-knowledge proofs with a prepared encoding of the claims for to fit the schema.[36]

In this model, actors are represented by a Decentralized Identifier (DID),[6] which is used to associate all their identity traits to them. This is a global and unique identifier that allows entities to be identified without the need of a centralized registry.

All these layers work independently of the architecture used to verify the information. The information regarding the issuers of the Verifiable Credentials, and available for all participants, can be as centralized or decentralized as desired.

One of the best ways of storing this information in a decentralized manner is by using Decentralized Ledger Technologies (DLT) or blockchain. It allows to register the issuers' information publicly in the ecosystem, making easier the verification of information and credentials. Since the information regarding issuers refers to legal entities or persons, there are not concerns regarding General Data Protection Regulation[8] and blockchain usage.[37]

By using the widely adopted[10] W3C data model for Verifiable Credentials[7] in the European Union,[21,22] Federated Data Spaces solve the issue of heterogeneity in shared data,[38] making easier to identify each data type, issuer, owner, in a uniform way.

Blockchain for identity governance

Usage of blockchain in identity solutions has been widely discussed[2] due to the concerns about the immutability of the information stored on it and GDPR compliance.[39] One of the main recommendations in enterprise environments is the usage of permissioned and private blockchains instead of public ones. This is due to the ability to have a wider control reach on the nodes' stored information, authentication, and identification of participants. These traits allow for easier GDPR compliance, allowing network administrators to manage the stored information and correct any necessary error.

Hyperledger Besu[30] is an open-source Ethereum client that allows to deploy and operate blockchains under controlled conditions for enterprise applications. Quorum Byzantine Fault Tolerant (QBFT)[31] consensus is the default protocol used on Hyperledger Besu permissioned blockchains. In this consensus there must be approved accounts that have the role of validators of the network. These are the ones that validate and approve transactions before including them in the next block. In order to do so, they must vote as a majority, two out of three, before the acceptance of a new transaction. Validators also vote for the inclusion or exclusion of others.

Nonetheless, GDPR does not apply to legal persons or entities,[8] allowing storage of information regarding them for public consultancy and validation of data origin and issuance.

European solutions, such as the European Blockchain Service Infrastructure,[23] adopt the usage of blockchain for identity, moreover in accordance with the W3C Verifiable Credentials model.[24]

This shows that there exists an interest in the synergy between blockchain technologies and Verifiable Credential models, allowing for transparency and decentralization of identity together.

EDC and trust extensions

Eclipse Data Components (EDC)[26] is a framework[25] for sharing data while maintaining sovereignty through a connector in Data Spaces. This connector complies with the proposed Gaia-X European framework,[13] making it suitable for solutions running in European territory.

Due to this, ECD has been widely adopted by different Data Spaces such as Eona-X,[15] a dataspace in the domain of mobility, transport and tourism, declared lighthouse project for GAIA-X; and Catena-X,[14] a dataspace for the automotive industry, which relies on both GAIA-X and the International Data Spaces Framework.

However, EDC is still under development, with iterative versions being published with added functionalities. This brings a few issues regarding backwards compatibility, because newer connectors do not guarantee correct functioning when interacting with previous versions. This is due to the added features and, therefore, updated data schemes and communication models.

Regardless of it, its usage is widely adopted, with most Data Spaces using one of the latest versions. Reasons include its interoperability with Verifiable Credentials and design compatible with the GAIA-X framework.

Summary of research gap

All the previously analyzed technologies and architectures are partially implemented by identity approaches. Public Key Infrastructure (PKI) is a component needed for decentralized identity, but by itself it cannot verify specific attributes of a credential or attestation. On the other hand, OAuth2 and OIDC are not decentralized, depending of a centralized entity to issue the tokens used by the actors to authenticate themselves. Self-Sovereign Identity (SSI) by itself is decentralized, but not the verification of the origin of each issued Verifiable Credential. For that, usage of Blockchain is needed in order to register each valid issuer. SSI's integration with EDC is, nonetheless, emerging, with newest versions of EDC making its implementation possible.

The proposed approach here uses Self-Sovereign Identity with blockchain technologies and usage of PKI and Decentralized Identifiers in order to create a decentralized identity model for Data Spaces. The Table 1 sumarizes the comparison between all the different identity approaches.

Problem statement and research questions

Problem statement

The evolution of federated data spaces has made trust a central architectural concern rather than a secondary operational feature. In these environments, independent organizations exchange data through connectors while preserving control over their assets, policies, and participation conditions. This model supports data sovereignty, but it also creates a demanding trust scenario: before a connector can disclose catalog entries, negotiate a contract, or initiate a transfer, it must determine whether the counterparty is a legitimate participant, whether the attributes it presents are valid for the intended interaction, and whether those attributes originate from issuers that are recognized as trustworthy within the ecosystem.

This problem cannot be reduced to simple endpoint authentication. In a federated data space, trust must cover at least four dimensions. First, the participant must have a verifiable identity that can be resolved and associated with valid service

Table 1. Summary of research gap.

Approach	Decentralized identity	Verifiable attributes	Shared governance	EDC integration	Operational evaluation
PKI	Partial	No	Limited	Yes	Common
Oauth2/OIDC	No/partial	Limited	Centralized/ federated	Yes	Common
SSI without blockchain	Yes	Yes	Off-chain	Emerging	Limited
Blockchain SSI	Yes	Yes	On-chain	Limited	Limited
Proposed approach	Yes	Yes	On-chain	Yes	Evaluated

endpoints. Second, the participant must be able to prove the attributes required for interaction, such as membership, permissions, or sector-specific affiliation. Third, the issuers of those claims must themselves be subject to governance, since cryptographic validity alone does not guarantee institutional legitimacy. Fourth, all trust-relevant decisions must remain auditable over time, so that changes in participation status, issuer authorization, or identity state can be inspected and justified afterwards.

The background discussed in Section 2 shows that existing approaches address these requirements only partially. As summarized in Table 1, PKI-based and OAuth2/OIDC-based solutions are operationally mature and can be integrated with connector ecosystems,[4] but they rely on centralized or federated trust anchors. As a result, trust ultimately depends on a limited set of authorities that control identity issuance and validation. This model is effective in bounded administrative environments, but it is less well suited to federated ecosystems in which multiple organizations must recognize each other without depending on a single identity provider or on a single institutional gatekeeper.

SSI-based approaches improve this situation by making identity and attributes portable, cryptographically verifiable, and more directly controlled by the participants themselves. Through decentralized identifiers and Verifiable Credentials,[6,7] SSI enables participants to prove who they are and which claims about them have been issued. However, SSI alone does not fully solve the trust problem of federated data spaces. In particular, it does not by itself answer who is authorized to register new identities, which issuers should be accepted by the ecosystem, how trust relationships are updated or revoked, or how these governance decisions can be shared transparently among participants. Without a governed trust layer, SSI may improve identity portability while leaving issuer recognition and participation control fragmented, bilateral, or implicitly centralized.

Blockchain-supported identity governance addresses part of this gap by introducing a shared and auditable substrate for role management, issuer authorization, and identity registration.[3] In such a model, trust-relevant state changes are no longer hidden inside a single organization, but recorded in a replicated and tamper-resistant infrastructure. Even so, a second gap remains: many blockchain-based proposals do not show clearly how governed trust decisions can be operationalized in real connector interactions. In other words, it is not sufficient to state that identities and roles are governed on chain; it must also be shown how this governance affects actual connector behavior during discovery, negotiation, and transfer.

This is precisely the point at which federated data spaces require a more complete solution. Trust must not only be defined at the governance level, but also enforced at the interaction level. A connector should be able to discover the counterparty through a verifiable identity, validate the presented credentials, determine whether their issuers are recognized, and apply these checks within the normal sequence of catalog access, contract negotiation, and transfer initiation. At the same time, the architecture should preserve a clear separation between infrequent governance-intensive operations, such as identity registration or role assignment, and recurrent runtime operations, such as DID resolution and credential verification.

Accordingly, the problem addressed in this paper is how to provide a decentralized, governed, and operationally usable trust architecture for federated data spaces that combines self-sovereign identity, Verifiable Credentials, shared governance of roles and issuers, interoperability with Eclipse Dataspace Components, and an experimental basis for evaluating its operational behavior. The goal is not simply to propose another identity model, but to define a trust architecture that can be embedded into connector-based data exchange and analyzed in terms of integration, performance, governance, and security.

Research objective and scope

Based on the previous problem statement, the objective of this paper is to define and evaluate a blockchain-governed SSI trust architecture for federated data spaces. More specifically, the paper aims to show how decentralized identifiers, Verifiable Credentials, permissioned blockchain governance, and connector-level trust enforcement can be combined into a coherent solution that is both technically interoperable with Eclipse Dataspace Components (EDC)[25] and meaningful from governance and security perspectives.

The scope of the paper is intentionally focused. It addresses the trust layer required for participant interaction in federated data spaces, with particular attention to identity registration, issuer recognition, credential-based authorization, and connector interoperability. In this sense, the work focuses on how trust is established, represented, governed, and enforced across the main stages of connector interaction. It does not attempt to redesign the Dataspace Protocol itself, replace the internal policy engine of the connector, or provide a complete production-scale governance framework for all possible data-space ecosystems.

Likewise, the paper does not treat blockchain as a storage layer for business data, nor does it address data-plane confidentiality or throughput as its main research target. Instead, the focus is on the trust and identity mechanisms that make governed participation possible. This analytical boundary is important because it allows the contribution of the work to be assessed precisely: the proposal is intended as a reusable trust layer for connector-based federated data spaces, rather than as a general-purpose redesign of the whole data-space stack.

Research questions

To address the identified problem within this scope, the paper is guided by the following research questions:

1. How can a blockchain-governed SSI model provide verifiable identity, trusted issuer control, and auditable governance for participants in a federated data space?

This question focuses on the conceptual trust model of the proposal. It concerns the combination of decentralized identifiers, Verifiable Credentials, issuer recognition, and permissioned blockchain governance as the basis for establishing trust among independent participants.

2. Which architectural components are required to realize this trust model while preserving a clear separation between governance, identity resolution, credential management, and connector operation?

This question addresses the architectural realization of the model. It is concerned with the structure of the solution, the roles of its main components, and the way in which governance-intensive and runtime-intensive functions are separated to keep the architecture both trustworthy and operationally usable.

3. How can the proposed trust model be integrated with Eclipse Dataspace Components (EDC)[25] so that trust verification becomes part of catalog discovery, contract negotiation, and transfer initiation?

This question focuses on connector interoperability. It examines how the trust layer can be embedded into the normal lifecycle of EDC-based interactions, ensuring that identity resolution, credential validation, and issuer trust are not external administrative checks but active parts of the connector workflow.

4. What operational, governance, and security implications emerge from the proposed approach when assessed through experimental evaluation and trust analysis?

This question concerns the evaluation of the proposal. It covers the practical effects of the architecture in terms of measurable overhead, separation between onboarding and runtime costs, governance properties, and the main trust and security implications that arise from the design.

These research questions are intentionally aligned with the structure of the remainder of the paper. Section 4 answers RQ1 by defining the blockchain-governed SSI trust model. Section 5 answers RQ2 by presenting the architecture that realizes that model. Section 6 addresses RQ3 through the integration with Eclipse Dataspace Components. Sections sec: methodology and 8 contribute to RQ4 through the experimental methodology and results, while Section 9 completes that analysis from the perspectives of security, governance, and trust. In this way, the research questions do not open a separate line of inquiry disconnected from the rest of the article; rather, they provide an explicit analytical structure for the contribution that is already developed in the subsequent sections.

Blockchain-Governed SSI trust model for federated data spaces

This section addresses RQ1 by defining the trust model that underpins the proposal. The objective is not yet to describe the technical implementation of the system, which is deferred to Section 5, but to establish the conceptual rules according to which trust is created, governed, and validated in the proposed federated data-space environment.

The model starts from a simple premise: in a federated data space, trust cannot be reduced to the existence of a valid network endpoint or to the presentation of a cryptographically well-formed credential. A participant must be trusted as an actor of the ecosystem, its attributes must be trusted as relevant and current for the interaction, and the issuers of those attributes must themselves be trusted according to a governance framework shared by the ecosystem. For this reason, the proposed model combines self-sovereign identity artifacts, Verifiable Credentials, and blockchain-governed trust state into a single decision framework.

Model foundations

The trust model is based on four design principles.

First, identity must be decentralized but verifiable. Each participant is represented through a decentralized identifier (DID)[6] that binds the participant to cryptographic material and service metadata through a DID Document. This allows connector discovery and participant authentication to be based on a portable and verifiable identity artifact rather than on static endpoint configuration or on the control of a central identity provider.

Second, attributes must be portable but not self-declared. The model uses Verifiable Credentials (VCs)[7] to represent membership, permissions, sector affiliation, or other ecosystem-relevant claims. This means that connectors do not rely on opaque bilateral assertions. Instead, they evaluate explicit credentials that can be verified cryptographically and interpreted according to the participation rules of the data space.

Third, trust must be governed rather than assumed. Even when a DID can be resolved and a credential can be verified cryptographically, the ecosystem still needs a mechanism to determine which identities may be registered, which issuers are recognized, and which privileged actions are allowed. In the proposed model, these decisions are anchored in a permissioned blockchain network[30] and expressed through smart-contract-governed state transitions.[5] This avoids concentrating trust in a single administrative intermediary while preserving a shared and auditable governance substrate.

Fourth, trust must be interaction-aware. A participant is not trusted in the abstract, but in relation to a concrete action within the data-space workflow. Trust must therefore be re-evaluated when a connector requests a catalog, initiates a contract negotiation, or attempts to start a transfer. This principle ensures that governed identity and credential validation remain part of the operational lifecycle of the data space rather than being confined to an isolated onboarding stage.

Taken together, these four principles define a trust model that is neither purely identity-centric nor purely governance-centric. Instead, it links identity, attributes, issuer legitimacy, and access conditions into a single decision process that can later be embedded into the connector interaction flow. Figure 1 shows this interaction.

Trust evaluation semantics

Under the proposed model, trust is expressed as a composite condition. Let p denote a participant and let a denote a specific interaction request within the data space, such as catalog access, contract negotiation, or transfer initiation. The participant is considered trustworthy for that interaction only if all relevant trust conditions are satisfied:

$$T(p,a) = V_{did}(p) \wedge V_{role}(p) \wedge V_{cred}(p,a) \wedge V_{iss}(p,a) \wedge V_{pol}(p,a)$$

In this formulation:

- $V_{did}(p)$ denotes that the participant's DID can be resolved correctly and that the associated DID Document is valid for use in the ecosystem.
- $V_{role}(p)$ denotes that the participant holds an active status within the governed trust domain, including any role required for participation.

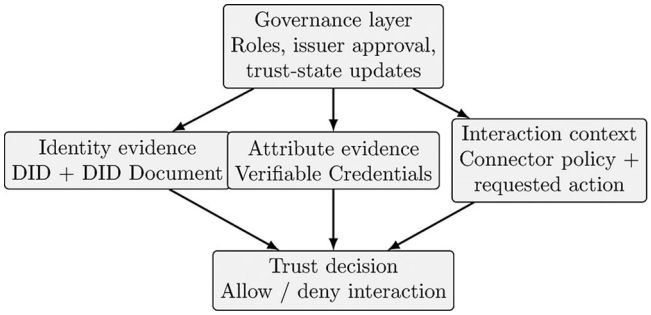

Figure 1. Conceptual trust model of the proposed blockchain-governed SSI approach. Governance defines which identities, issuers, and conditions may be accepted, while runtime trust decisions combine identity evidence, attribute evidence, and policy context.

- $V_{cred}(p,a)$ denotes that the participant can present the Verifiable Credentials required for the requested interaction.

- $V_{iss}(p,a)$ denotes that the issuer of those credentials is recognized as trustworthy according to the governance framework of the ecosystem.

- $V_{pol}(p,a)$ denotes that the requested interaction satisfies the connector-level policy conditions applicable to the asset or action in question.

This expression makes explicit a distinction that is central to the model: cryptographic validity is necessary but not sufficient for trust. A credential may be well formed and correctly signed, yet still be unacceptable if its issuer is not recognized by the ecosystem. Likewise, a valid DID is not sufficient if the participant does not hold the credentials required for the intended interaction. Trust therefore emerges from the conjunction of identity, governance, credential legitimacy, and policy satisfaction.

Governance roles and trust domains

The proposed model distinguishes between the actors that govern trust and the actors that consume trust during data exchange. This distinction is important in federated data spaces because the entity that operates infrastructure should not necessarily be the same as the one that issues credentials, and neither should automatically coincide with the one that defines participation rules.

At the conceptual level, the trust domain is structured around four roles: governance authority, network operator, trusted issuer, and operational participant. Their functions are summarized in Table 2.

The governance authority is responsible for defining the participation framework of the data space and for approving trust-sensitive operations such as identity admission, issuer recognition, and major changes to the trust infrastructure. Its role is not to authenticate every interaction directly, but to provide institutional coherence to the ecosystem and to ensure that trust relationships evolve according to explicit rules.

The network operators maintain the permissioned blockchain on which governance state is anchored. Their role is infrastructural rather than semantic: they preserve the integrity and continuity of the ledger, but they do not automatically gain the right to issue credentials or redefine trust relationships. This separation reduces concentration of power and is consistent with the governance logic expected in data-space ecosystems.[12,13]

The trusted issuers are the entities allowed to attest claims about participants or connectors through Verifiable Credentials. In the proposed model, recognition of issuers is not left to ad hoc bilateral trust. Instead, it is governed at the ecosystem level, so that credential acceptance depends not only on cryptographic validity but also on the recognized status of the issuer.

Finally, the operational participants are the entities that exchange data through connectors. Their effective trust status depends on the combination of their decentralized identity, their current credentials, the recognition of the corresponding issuers, and the policy context of the requested interaction. This makes trust contextual and governed at the same time.

Table 2. Governance roles in the proposed blockchain-governed SSI trust model.

Role	Main responsibility	Trust relevance
Governance authority	Defines participation rules and authorizes trust-sensitive changes	Provides institutional legitimacy to identity registration, issuer approval, and governance evolution
Network operator	Maintains the permissioned blockchain infrastructure and consensus process	Preserves the availability and integrity of the shared governance substrate
Trusted issuer	Issues Verifiable Credentials about participants or connectors	Certifies ecosystem-relevant claims that can later be evaluated in connector interactions
Operational participant	Uses a connector to request or provide data within the ecosystem	Consumes governed trust by presenting identity and credentials during interaction

This role separation also supports regulatory coherence. Since the model does not require storing personal business data on-chain and instead anchors governance-relevant state while keeping credentials as separate verifiable artifacts, it aligns with the principles of data minimization and controlled trust representation associated with the GDPR[8] and with the European trust-services perspective reflected in eIDAS.[9]

Trust lifecycle

The proposed trust model spans the full lifecycle of participation in the data space, from admission to runtime interaction and later trust-state changes.

During onboarding, a participant first enters the trust domain through a governed admission process. This stage establishes the participant's identity in the ecosystem, associates it with an appropriate role, and creates the basis on which further claims may be issued. At this point, the participant does not yet obtain unconditional access to all possible interactions; rather, it becomes a recognized subject of governed trust.

Once admitted, the participant may obtain one or more Verifiable Credentials from trusted issuers. These credentials capture the claims that are relevant for future interactions, such as membership in the ecosystem, permission to operate in a specific role, or affiliation with a given sector. Because credentials are externalized from the blockchain itself, the model preserves portability and selective disclosure while still relying on the governed trust domain to determine which issuers are acceptable.

During runtime interaction, trust is evaluated in relation to specific connector operations. When one connector interacts with another, it does not rely solely on static configuration or on a once-issued admission token. Instead, it must be able to resolve the counterparty's identity, verify the credentials presented, determine whether the issuer is recognized, and evaluate the request against the relevant policy conditions. This is the operational expression of Eq. 1: trust is checked when it becomes relevant for disclosure, negotiation, and transfer decisions.

The model also incorporates a maintenance and revocation dimension. Participation status may change over time; issuers may lose their recognized status; DID metadata may need to be updated; and credentials that were once acceptable may cease to be so under new governance conditions. For this reason, trust is not treated as immutable. Governance state can be updated through the permissioned blockchain, and future runtime verifications must reflect those changes. In this way, the model supports both persistence and controlled evolution of trust relationships.

Expected properties of the model

From the previous definition, four expected properties follow.

First, the model is designed to support decentralized identity with shared governance. Identity is not controlled by a single central provider, yet the ecosystem still retains explicit control over admission and issuer recognition.

Second, it is designed to support credential-based trust with issuer accountability. Claims are portable and verifiable, but their acceptance is conditioned by the governed legitimacy of the issuers.

Third, it is designed to support auditable evolution of trust. Since identity registration, role assignment, and issuer authorization are tied to a shared governance substrate, trust relationships can evolve without becoming opaque or purely bilateral.

Fourth, it is designed to support operational interoperability. The model does not define trust as a separate administrative layer detached from data exchange, but as a decision framework that can be enforced during the main stages of connector interaction.

Architecture

This section addresses RQ2 by translating that model into a concrete architectural organization suitable for federated data spaces. The focus here is on the structural realization of the proposal: which components are required, how responsibilities are separated, which elements belong to the shared trust domain, and which remain under the control of each participant.

The architecture is designed around a central tension already identified in Section 3. On the one hand, trust-relevant decisions such as identity registration, issuer approval, and role assignment must be governed, auditable, and shared across the ecosystem. On the other hand, the recurrent runtime operations of a data space, such as service discovery,

credential verification, catalog access, contract negotiation, and transfer initiation, must remain lightweight enough to be used in practice. The proposed architecture addresses this tension through a separation between governance-intensive functions and interaction-intensive functions, while preserving explicit interfaces between both.

Architectural decomposition

The architecture is decomposed into two broad scopes: a shared trust domain and a set of participant-local connector domains. This distinction is fundamental to the proposal because it preserves the autonomy of each participant while still enabling the ecosystem to share a common governance substrate.

The shared trust domain contains the components that define and preserve the common trust state of the federated data space. These include the permissioned blockchain network, the smart contracts governing participation, and the credential issuer service responsible for producing Verifiable Credentials[7] accepted in the ecosystem. These elements are shared because they define facts that must be consistently recognized by all participating actors.

The participant-local connector domain contains the components operated by each participant to consume that shared trust state during actual connector interactions. These include the EDC connector itself, the local identity and credential support services, and the identity proxy instance that mediates access to the decentralized trust infrastructure. These components are kept local because they belong to the operational boundary of each participant and must remain under its direct control.

This decomposition ensures that the architecture is neither fully centralized nor fully replicated without structure. Governance-relevant state is shared where common recognition is required, whereas operational interaction remains distributed across the participants.

Core components

Although Figure 2 provides the structural view of the system, the architecture is better understood by clarifying the role of its main components. Table 3 summarizes them.

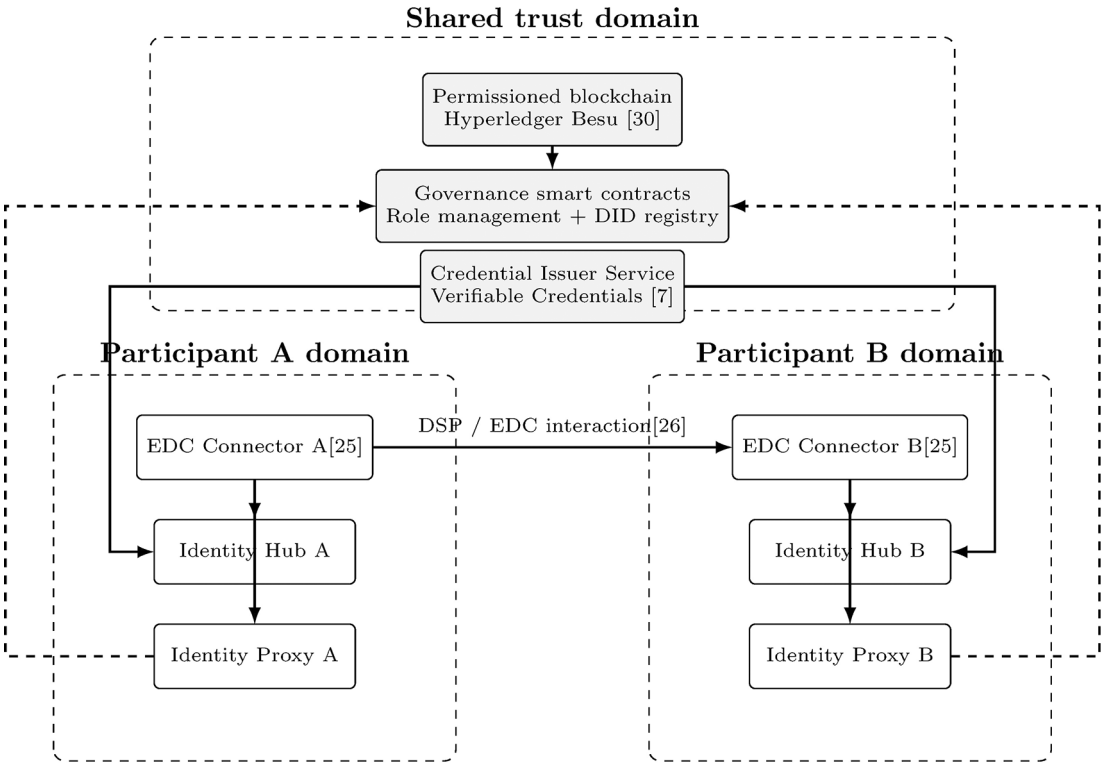

Figure 2. Static overview of the proposed architecture. Shared governance and credential issuance are maintained in a common trust domain, while each participant operates a local connector domain containing the EDC connector, credential support, and identity-proxy mediation services.

Table 3. Main architectural components of the proposed solution.

Component	Scope	Main responsibility
Permissioned blockchain + governance contracts	Shared	Maintains governed trust state, including identity registration, role assignment, and authorization of trust-relevant actions
Credential Issuer Service	Shared	Issues Verifiable Credentials representing ecosystem-relevant claims such as membership, permissions, or sector affiliation
Identity Proxy	Participant-local	Resolves DIDs, retrieves DID Documents, mediates trust-related queries, and shields the connector runtime from infrastructure-specific complexity
Local credential store	Participant-local	Stores and provides the credentials required by the connector runtime for decentralized identity and trust interactions
Connector runtime	Participant-local	Executes the data-space workflow and invokes trust services during discovery, negotiation, and transfer initiation
Audit/logging facilities	Cross-cutting	Preserve traceability of trust-relevant events and support later governance and security analysis

The permissioned blockchain network provides the shared trust anchor for the ecosystem. In the proposed architecture, its role is not to serve as a general-purpose data store, but to host the governance state that must remain visible, controlled, and auditable across participants. This includes identity registration and the state transitions associated with trust-relevant roles and permissions. By relying on a permissioned Hyperledger Besu deployment configured for governed participation,[30] the architecture ensures that updates to the trust domain are not left to unilateral local control.

The governance smart contracts[40] express the participation logic of the ecosystem in an enforceable form.[5] At a minimum, the architecture assumes a DID registration capability and a role-management capability, so that the ecosystem can determine who may register identities, which participants may exercise privileged actions, and how those permissions evolve over time. This makes trust-state transitions explicit and auditable instead of administrative and opaque.

The Credential Issuer Service is responsible for producing the Verifiable Credentials used in subsequent connector interactions. It occupies an intermediate architectural position: it is external to the connector runtime, yet it is not merely an administrative add-on. Through it, governance decisions become usable identity artifacts, since membership, authorization, or sector-specific claims can be materialized as credentials that participants later present to each other. The architecture intentionally keeps these credentials off-chain. This reduces ledger exposure, avoids unnecessary replication of credential payloads, and preserves the selective-disclosure logic associated with SSI.[7]

The Identity Proxy is one of the key architectural elements of the proposal. It acts as a trust-mediation service between the connector runtime and the decentralized trust infrastructure. Instead of embedding blockchain-specific logic directly into the connector, the architecture delegates DID resolution, DID Document retrieval, and related trust validation tasks to the proxy. This choice has two benefits. First, it reduces the degree of coupling between the connector and the underlying identity infrastructure. Second, it centralizes resolution and validation logic in a component that can be deployed, monitored, and evolved independently. In practical terms, the proxy functions as the local gateway through which each participant accesses governed identity state.

The Identity Hub complements this role at the credential layer. In the implemented EDC-compatible design, decentralized identity and trust depend not only on resolving counterpart identities, but also on storing and providing the credentials that the local connector must use during interaction. The Identity Hub fulfills this purpose by keeping the credentials accessible to the connector runtime while preserving a clean separation between connector logic and credential custody.

Finally, the EDC connector and trust extension form the operational entry point of the architecture. The connector remains responsible for the native data-space workflow, including catalog access, policy-aware contract negotiation, and transfer initiation.[25,26] The trust extension does not replace that workflow. Instead, it enriches it by invoking the identity and credential services when trust-relevant decisions must be made. This preserves compatibility with the connector model while embedding governed trust verification into the normal lifecycle of interaction.

Layer responsibilities and separation of concerns

A central architectural property of the proposal is the explicit separation of concerns across layers. This separation is not only a software-engineering choice; it is necessary to preserve the trust semantics defined in Section 4.

The governance layer is responsible for deciding which identities, roles, and issuers may be recognized by the ecosystem. Its outputs are durable and comparatively slow-moving. Identity registration, role assignment, or issuer approval are not expected to occur at the same frequency as catalog queries or negotiations. For this reason, the architecture isolates them in a shared substrate where consistency and auditability take precedence over minimum latency.

The identity and credential services layer translates that governed state into artifacts and queries that can be consumed operationally. The identity proxy transforms on-chain or governed identity state into resolvable DID-based information; the credential issuer transforms governed recognition into portable credentials; and the Identity Hub makes those credentials available to the participant-side runtime. This layer acts as the bridge between governed trust and operational trust.

The connector operation layer remains focused on the federated data exchange process itself. This is important because the architecture does not aim to redefine the connector as an identity platform. Its role is still to manage catalog requests, negotiation, and transfer according to EDC semantics. Trust services therefore remain externalized and are consulted where necessary, but they do not collapse into the core data-exchange logic.

This separation brings a further benefit: it allows the architecture to keep onboarding-intensive operations distinct from runtime interaction-intensive operations. Onboarding requires governed admission, identity creation, and issuance of initial credentials. Runtime interaction, by contrast, depends mainly on DID resolution, credential retrieval, and verification. The architecture is intentionally organized so that the heavier governance mechanisms are not paid in full at every interaction, even though their effects continue to shape every trust decision.

Deployment scopes and trust boundaries

Beyond functional decomposition, the architecture also defines clear deployment scopes and trust boundaries.

The first trust boundary lies between the shared trust domain and the participant-local connector domain. The shared trust domain is responsible for common recognition, but it is not allowed to absorb all participant logic. This preserves participant autonomy and avoids reintroducing a de facto centralized connector architecture under the guise of a decentralized identity system.

The second trust boundary lies between the connector runtime and the identity infrastructure. Connectors do not query the blockchain directly and do not assume direct control over issuer logic. Instead, they rely on mediated services. This boundary reduces complexity inside the connector runtime, limits the impact of infrastructure evolution on connector behavior, and allows identity-specific validation rules to be encapsulated in dedicated services.

The third trust boundary concerns credential custody. Credentials are not embedded permanently into the connector implementation or anchored in full on-chain form. They remain managed through dedicated identity services and are used by the connector when required. This supports a cleaner separation between identity evidence and data-exchange logic, while remaining aligned with the SSI principle that identity artifacts should remain portable and selectively disclosable.

These boundaries also clarify what the architecture does not assume. It does not assume that the blockchain alone is sufficient to operationalize trust, nor that credentials alone are sufficient to define issuer legitimacy, nor that connector interoperability alone is sufficient to create governed participation. Instead, trust emerges from the interaction between these bounded components.

Architectural rationale

The proposed architecture is shaped by four practical design choices.

First, it favors minimal blockchain exposure. Only governance-relevant state is anchored in the shared ledger, while credentials and interaction payloads remain outside it. This keeps the blockchain focused on shared trust coordination rather than turning it into a general-purpose storage layer.

Second, it favors connector decoupling. By using the identity proxy as a mediation service, the architecture avoids embedding ledger-specific identity resolution and validation logic directly into the connector core. This makes the overall solution more maintainable and more adaptable to future evolution of the trust infrastructure.

Third, it favors participant symmetry. Each participant domain follows the same local architectural pattern: connector, credential support, and identity-proxy mediation. This symmetry is important in federated data spaces, where trust should not depend on asymmetric connector roles except where the interaction itself requires it.

Fourth, it favors auditability without loss of autonomy. The ecosystem shares a common governance substrate, but each participant retains operational control over its own connector boundary and local trust-service deployment. This combination reflects the balance between common trust rules and local sovereignty that characterizes federated data spaces.[12,13]

In summary, the architecture realizes the trust model of Section 4 through a layered and bounded organization in which shared governance, portable credentials, mediated identity resolution, and connector-level enforcement remain clearly separated yet operationally linked. On this basis, Section 6 details how these architectural elements are integrated into Eclipse Dataspace Components and how trust checks are embedded into the connector interaction lifecycle.

Integration with eclipse dataspace components

EDC connector trust requirements

To initiate an interaction with another connector, a participant must first know the counterparty's DID.[6] By resolving this DID, it can retrieve the corresponding DID document and obtain the DSP endpoint[26] required to request the catalog through the EDC control plane.[27] This discovery step is essential because it links the service endpoint to a verifiable identity, thereby reducing the risk of relying on unauthenticated connection metadata.

In addition to endpoint discovery, both connectors must possess the cryptographic keys and valid credentials[7] required to authenticate themselves and prove their role within the data space. The proposed solution therefore requires a mechanism for determining which credential issuers are trusted. In this model, this function is performed by the identity proxy,[29] which supports issuer validation and credential verification before the interaction progresses to subsequent stages of the connector workflow.[27]

These trust checks are not limited to the initial connection phase. DID resolution and connector validation are carried out throughout the main stages of the interaction,[28,27] namely during the catalog request, the initiation of negotiation, and the initiation of the transfer process defined by DSP.[26] Reapplying these controls at each stage helps ensure that access decisions remain aligned with the connector's current identity state, credentials, and authorization status.

DID resolution extension

In the proposed integration, EDC obtains the counterparty's DID[6] as part of the information required to initiate the catalog request. Once this identifier is known, the connector resolves it through the identity proxy[29] in order to retrieve the associated DID Document. This approach avoids reliance on manually configured network metadata and ties connector discovery to a verifiable identity artifact.

The DSP endpoint is extracted from the service entries contained in the DID Document.[6] More specifically, the document includes the service definition through which Dataspace Protocol interactions are exposed,[26] allowing the requesting connector to determine the correct DSP URL for the catalog request and for the subsequent negotiation and transfer stages in the control plane.[27]

Validation of the DID Document[6] is performed during the same resolution process. The identity proxy[29] verifies that the DID can be resolved correctly, that the returned document is well formed, and that its contents are consistent with the trust layer. Symmetrically, when the receiving connector processes the catalog request, it extracts the sender's DID from the presented Verifiable Credentials,[7] resolves it, and validates the corresponding DID Document before continuing the interaction.

The identity proxy therefore acts as the mediation layer between EDC[25] and the decentralized trust infrastructure. It abstracts the complexity of DID resolution, centralizes validation logic, and provides a stable interface for retrieving trusted identifiers and service endpoints within the connector control plane.[27] In the proposed solution, the connector communicates with the identity proxy through an EDC connector extension, which encapsulates the required resolution and validation calls within the connector runtime. As a result, connectors can enforce trust checks without embedding blockchain-specific resolution and verification mechanisms directly into the core connector logic.

Onboarding flow

Before participating in runtime data exchange, each connector must complete an onboarding process that establishes its initial trust status within the data space. As shown in Figure 3, the process begins when the participant operator registers the connector with the governance layer, which records the corresponding decentralized identity[6] and assigns the appropriate role within the ecosystem through governed contract logic.[5] Once registration is completed, the operator requests a participation credential from a trusted issuer, which issues a Verifiable Credential[7] that the connector can later present in subsequent interactions to prove its membership and authorized attributes.

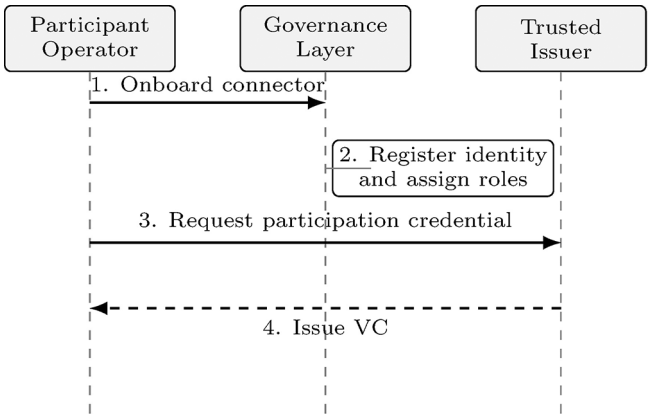

Figure 3. Operational sequence for connector onboarding.

Data exchange flow

The data exchange flow begins when the consumer identifies the provider's DID[6] and resolves it through the identity proxy. The proxy returns an adapted DID Document, from which the consumer extracts the DSP endpoint[26] required to interact with the provider connector. Using this endpoint, the EDC connector initiates the catalog request and subsequently starts the contract negotiation process in the control plane.[27] This sequence ensures that all interactions are derived from verifiable identity data rather than static or manually configured endpoints.

Before the transfer process is executed, the provider side performs validation of identity, credentials,[7] and applicable policies, typically involving the governance layer and the corresponding control-plane entities.[28,27] Only if these checks succeed does the interaction proceed to the data transfer stage. Throughout the process, relevant events can be recorded or audited, either within dedicated components such as an audit or logging layer or within the corresponding functional layer responsible for each step. This approach maintains traceability while preserving the separation of concerns across the proposed solution. Figure 4 shows this flow.

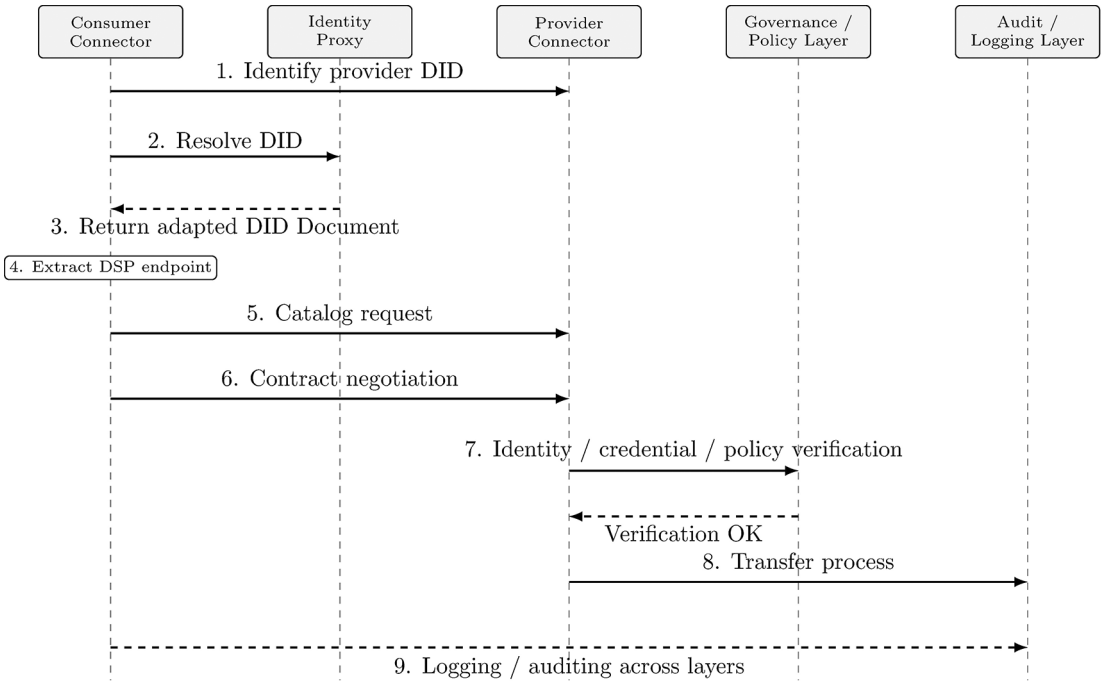

Figure 4. End-to-end interaction flow between consumer and provider connectors.

Credential-based access control

Verifiable Credentials[7] are used both to demonstrate membership in the data space and to prove that a participant holds the permissions required to operate within it. In addition, they can be used to attest sector-specific attributes, enabling participants to discover and request datasets that are relevant to their own domain. Credentials related to membership and operational authorization, as handled by the identity layer,[29] allow participants to access other connectors' catalogs and to initiate negotiation and transfer processes in the connector control plane.[27] Sector-related credentials, in turn, make it possible to restrict catalog visibility so that participants are only presented with datasets associated with their corresponding sector.

This distinction supports a more fine-grained access-control model in which eligibility to participate in the ecosystem and eligibility to access specific categories of data can be evaluated separately. In practice, the connector can therefore assess not only whether the counterparty is a trusted member of the data space, but also whether it satisfies the domain-specific conditions associated with a given dataset or policy.[28,26] As a result, Verifiable Credentials[7] provide a flexible mechanism for enforcing both ecosystem-level trust requirements and more specific data-sharing constraints, while preserving interoperability and reducing unnecessary exposure of information.

Sector-specific verification is enforced through the combined use of Verifiable Credentials[7] and policy rules. Participants that wish to prove their affiliation with a given sector must first obtain the corresponding credential from an authorized issuer. Likewise, participants offering sector-specific data may configure policies that declare the relevant sector category associated with those assets and define the conditions under which they may be disclosed. When one connector requests the catalog of another, and the provider has defined a sector-based policy, the provider evaluates the requester's Verifiable Credential to determine whether the declared sector matches the one required by the policy within the control-plane decision flow.[27,28] This evaluation can be performed before catalog entries are disclosed, thereby ensuring that visibility decisions are aligned with the provider's data-sharing constraints from the outset of the interaction. If the validation is successful, the requester is allowed to view the assets configured for that sector; otherwise, those assets remain hidden. In this way, sector-specific credentials support selective disclosure and policy-driven access control without requiring the provider to expose irrelevant or unauthorized resources.

Experimental methodology

For evaluation purposes, the proposed solution presented in the previous sections was materialized in the development of an implementation called IDBLOCK.[11] In the remainder of the experimental sections, IDBLOCK therefore refers to this concrete implementation, while the term proposed solution denotes the general architectural model.

Experimental setup

The experimental environment was deployed on a single machine equipped with an 14th-generation Intel Core i9 processor and 32 GB of RAM. All components were executed locally, with both EDC connectors running on the same host and communicating through localhost. The blockchain layer was implemented using a single Hyperledger Besu node deployed in a Docker container and configured as a private network with chain ID 1337. Although the consensus mechanism was defined as QBFT,[31] the setup included only one validator node; consequently, fault tolerance was not exercised and consensus effectively reduced to a local block-validation process. The configuration used a block period of 2 seconds, an epoch length of 30000, and a request timeout of 4 seconds. Gas-pricing constraints were eliminated by setting the base fee to zero.

At the application layer, two EDC connectors[25] were deployed in separate containers, representing the consumer and provider roles. The IDBLOCK deployment also included several auxiliary services: a DID registry implemented as a smart contract on Besu and extended with traceability functionality; one identity proxy instance associated with each connector, responsible for DID resolution and validation of DIDs[6]; and a centralized credential issuer service hosted within an internal network for credential handling.[7] During the data-exchange flow, the consumer connector resolved the provider's DID through its proxy, obtained the DSP endpoint, and initiated the catalog and negotiation processes via EDC control-plane operations.[27] Identity, credential, and policy verification were enforced before the transfer process was executed, and relevant events could be logged or audited within the corresponding layers. Logging was enabled at both the connector and proxy levels in order to capture interaction steps and timing information for subsequent analysis.

Evaluated operations

The evaluation focuses on the end-to-end latency introduced by the trust and identity functions that complement the connector workflow. The operations selected for measurement cover both the onboarding phase and the recurrent exchange phase. In particular, the experiments measure: (i) DID registration latency, corresponding to the time required to anchor a new participant identifier on the blockchain; (ii) DID update latency, corresponding to modifications of an already registered DID Document;[6] (iii) DID resolution latency through the identity proxy; (iv) Verifiable Credential

issuance latency;[7] (v) credential delivery and verification latency, whenever the full issuance-to-validation flow is exercised; (vi) EDC catalog request latency with and without DID resolution; (vii) contract negotiation latency with and without the SSI-based verification layer; and (viii) transfer initialization latency after a successful negotiation in the connector control plane;[27]

Workloads and scenarios

The experimental campaign is structured around a set of scenarios designed to isolate the cost of each architectural layer. Scenario S1 establishes a baseline by measuring the native EDC interaction[27] without SSI support and without blockchain-based identity resolution. Scenario S2 introduces DID resolution[6] through the proxy while keeping the remaining EDC flow unchanged, thereby capturing the cost of decentralized identifier lookup. Scenario S3 extends this configuration with Verifiable Credential validation,[7] allowing the overhead of identity and authorization checks to be assessed jointly through the corresponding control-plane entities.[28] Scenario S4 evaluates the onboarding phase, including DID registration and credential issuance for a new participant. Finally, Scenario S5 measures repeated catalog requests and contract negotiations after onboarding has already been completed, so that any caching effects present in the connector or proxy layers can be observed.

Each scenario will be executed through an automated script that invokes the target operation repeatedly under controlled conditions. For every measured operation, the script performs 1000 repetitions[32] in order to obtain a sufficiently large sample for latency distribution analysis and for comparison across scenarios.

Metrics and measurement procedure

Latency is measured from the instant a client request is emitted to the instant the corresponding operation is completed and a response or confirmation is received. For blockchain-backed operations, this interval includes submission and confirmation up to the point at which the transaction can be considered final within the single-node Besu deployment.[31] For connector interactions, the interval includes the complete request-response path across the involved services, including proxy resolution and credential validation whenever those steps are enabled in the scenario under study in the control plane.[27]

To reduce bias caused by initialization effects, each test includes a short warm-up phase[32] before the measured iterations begin. After warm-up, the automated script records 100 observations per operation and scenario. The reported results will include average latency, median latency, 95th percentile, 99th percentile, and standard deviation. When applicable, measurements will distinguish between cold and warm executions, particularly for DID resolution and other requests that may benefit from connector- or proxy-level caching.

In order to reduce experimental noise, all components are executed on the same host, background activity is minimized as far as possible, and the measurement procedure is applied sequentially and consistently across scenarios. The start and end points of each measurement are kept fixed for a given operation so that comparisons reflect architectural differences rather than changes in instrumentation.[32] This makes it possible to separate the cost of the SSI and blockchain layers from the baseline connector behavior.

Reproducibility considerations

Reproducibility is supported by keeping the experimental configuration explicit and self-contained. The connector configuration, Besu network parameters, identity proxy settings, and credential-service setup are all defined in the deployment artifacts used for the evaluation. The execution workflow is driven by scripts that automate the invocation of each scenario and collect the resulting measurements in a consistent format. Containerized deployment descriptors, such as Docker Compose manifests, can therefore be used to recreate the same local environment and to rerun the experiments with equivalent settings.

Nevertheless, the results must be interpreted in light of the limitations of the testbed. Because all components run on a single machine and the blockchain network contains only one validator node, the evaluation emphasizes functional overhead and relative latency differences rather than distributed-network behavior or Byzantine fault tolerance under realistic multi-node conditions. For the same reason, measurements related to caching, local resource contention, and container scheduling should be understood as environment-dependent and should be reported together with the configuration used to obtain them.

Experimental results

DID registry and resolution performance

Table 4 summarizes the latency observed for the main DID-registry and DID-resolution operations. DID registration and DID update exhibit very similar behavior, with average latencies of 1998.29 ms and 2000.93 ms, respectively. This result

Table 4. Latency summary for DID registry and resolution operations. All values are reported in milliseconds.

Operation	Avg	Median	p95	p99
DID registration	1998.29	2026.74	2038.72	2041.27
DID update	2000.93	2031.38	2044.54	2047.12
DID resolution (cold)	2.59	2.10	4.62	5.59

is expected because both operations are executed as on-chain transactions against the DID Registry contract and therefore must wait for block inclusion and confirmation before completion. By contrast, cold DID resolution remains substantially faster, with an average latency of 2.59 ms and a p99 of 5.59 ms, since it is handled through the identity proxy and does not require writing state to the blockchain.

The observed range of the on-chain operations, from approximately 1.02 s to 2.05 s, is consistent with the 2 s block period configured for the single-node QBFT network.[31] In this environment, blockchain finality is effectively determined by the point at which the transaction is included in the next block, which means that the dominant share of DID registration and DID update latency is attributable to block finality rather than to smart-contract execution itself.[5]

From a system perspective, these results reveal a clear separation between infrequent but more expensive on-chain actions and lightweight runtime resolution. DID registration and DID update[6] are expected to occur mainly during onboarding or when a participant changes its identity metadata, so their higher cost has limited impact on the steady-state data-exchange path. DID resolution, on the other hand, is already low-latency in the cold path and can be further optimized through caching in the proxy or connector layers.[27] This suggests that the SSI-based trust layer introduces a measurable cost for state-changing operations while preserving efficient read access for recurrent connector workflows.

Credential issuance and verification

Table 5 reports the latency measured for the main credential-related operations. VC issuance is the lightest operation in the evaluated setup, with an average latency of 0.94 ms and a p99 of 1.43 ms, indicating that credential generation and signing introduce only a negligible overhead when executed in isolation. VC delivery is the dominant off-chain cost, with an average latency of 18.57 ms and a p99 of 28.64 ms, which suggests that transport and service-to-service exchange account for a larger share of the total delay than credential creation itself. VC verification remains moderate, with an average latency of 7.94 ms and a median of 6.47 ms, making it more expensive than issuance but still substantially cheaper than end-to-end credential delivery.

These results show that the credential layer does not introduce a uniform cost across all operations. Issuance is essentially negligible in comparison with the other stages, whereas delivery captures the largest end-to-end overhead in the measured path. Verification occupies an intermediate position: it is clearly more expensive than issuance because it includes cryptographic checks and validation logic for Verifiable Credentials,[7] but it remains well below the latency observed for credential transport between services mediated by the identity layer.[29]

Impact on EDC catalog and negotiation flows

Table 6 compares the average latency of the main EDC interaction steps in a baseline connector configuration and in the trust-enhanced deployment with IDBLOCK. The largest impact is observed in catalog requests: the average latency increases from 4.61 ms without SSI to 134.65 ms with IDBLOCK, which corresponds to an absolute overhead of 130.04 ms and a relative increase of 2820.82%. This behavior is consistent with the fact that catalog disclosure is the first trust-sensitive step in the interaction and therefore accumulates DID resolution, credential validation, and policy checks before assets are exposed.

Table 5. Latency summary for credential issuance, delivery, and verification. All values are reported in milliseconds.

Operation	Avg	Median	p95	p99
VC issuance	0.94	0.62	0.94	1.43
VC delivery	18.57	17.60	23.33	28.64
VC verification	7.94	6.47	10.10	12.99

Table 6. Average latency of EDC operations without SSI and with IDBLOCK. All values are reported in milliseconds.

EDC operation	Baseline	With IDBLOCK	Overhead
Catalog request	4.61	134.65	+130.04 (+2820.82%)
Contract negotiation	2.43	14.97	+12.54 (+516.05%)
Transfer initialization	2.52	19.09	+16.57 (+657.54%)

Contract negotiation also incurs a measurable slowdown, increasing from 2.43 ms to 14.97 ms on average, while transfer initialization rises from 2.52 ms to 19.09 ms. In relative terms, these increments appear large because the baseline connector operations are extremely lightweight; however, in absolute terms they remain modest in the local deployment, staying below 20 ms on average. This indicates that, once the interaction progresses beyond the initial discovery phase, the SSI-enabled trust checks add overhead but do not dominate the overall latency budget of the EDC workflow defined by DSP and the corresponding control-plane entities.[26,27,28]

The percentile values reinforce the same interpretation. For catalog requests, the p99 grows from 7.28 ms in the baseline case to 219.61 ms with IDBLOCK, confirming that trust-aware discovery is the most expensive path. By contrast, negotiation and transfer initialization remain comparatively stable even with the trust layer enabled, with p99 values of 23.34 ms and 33.03 ms, respectively. Overall, these results suggest that IDBLOCK concentrates most of its cost in the steps where identity[6] and authorization through Verifiable Credentials[7] must be established, while subsequent connector operations retain relatively low absolute latency in the control plane.[27]

Scalability analysis

The present deployment does not constitute a full scalability benchmark, since the experiments were executed with only two participants represented by two EDC connectors, each associated with a single DID, and a blockchain layer composed of one Besu node. As a consequence, the reported measurements characterize functional overhead and local latency under a minimal deployment, but they do not yet provide a quantitative characterization of behavior as the number of participants, DIDs, connectors, or validators increases. This limitation is especially relevant when interpreting metrics such as latency under concurrency, DID-resolution throughput, proxy saturation, and blockchain-node saturation, which were not stressed in the current setup.

From an architectural perspective, scalability is expected to depend on three main factors. First, increasing the number of participants and DIDs[6] enlarges the identity state maintained by the registry and increases the number of lookups that may be requested through the proxy. Second, increasing the number of connectors and concurrent resolution requests places pressure on the identity-proxy layer, which is the component most directly exposed to saturation in read-dominated workloads. In such scenarios, latency under concurrency and throughput of DID resolution become the most relevant indicators, especially because resolution is part of the critical path for trust-aware catalog access.[27] At the same time, this path may benefit from caching, meaning that scalability is likely to depend not only on the raw number of requests but also on the ratio between cold and warm resolutions.

The blockchain layer introduces a separate scaling dimension. With a single Besu validator, the current experiments do not exercise the effect of consensus across multiple nodes, so the impact of the validator set size remains outside the scope of the present measurements. In a larger QBFT deployment,[31] state-changing operations such as DID registration and DID update would be expected to incur additional latency due to inter-validator coordination and quorum formation, whereas read-oriented DID resolution would be affected more by RPC contention and node load than by consensus itself. For this reason, a complete scalability evaluation should vary the number of participants, DIDs, connectors, concurrent resolutions, and Besu validators, and should explicitly measure latency under concurrency, DID-resolution throughput, proxy saturation, blockchain-node saturation, and the additional overhead introduced by multi-validator consensus.

Summary of findings

Taken together, the experimental results show a clear distinction between infrequent onboarding operations and recurrent runtime operations. The highest costs are concentrated in onboarding-related actions, particularly DID registration and DID update, whose latency is dominated by on-chain finality in the Besu network. By contrast, operations such as DID resolution, credential issuance, and credential verification remain substantially lighter, which indicates that the trust layer does not impose the same performance penalty across all interaction stages.

Within the recurrent exchange path, DID resolution introduces an additional but moderate overhead in absolute terms, especially when compared with the cost of state-changing on-chain transactions. This overhead becomes visible in trust-aware catalog and negotiation flows, where identity resolution and credential checks are part of the critical path. Even so, the measured latencies remain acceptable for federated data-space scenarios in which strong trust guarantees, controlled participation, and policy-aware disclosure are more important than minimizing latency to the lowest possible level.

More broadly, the evaluation supports one of the central design goals of the proposed solution: separating frequent operational interactions from governance-intensive actions. Day-to-day connector operations can rely on relatively lightweight resolution and verification steps, while heavier governance operations such as identity registration, updates, role management, and issuer control are confined to less frequent events. This separation allows the system to preserve strong trust and auditability properties without forcing every interaction to pay the full cost of governance mechanisms.

Security, governance and trust analysis

Trust properties

The proposed solution provides verifiable identity by assigning each connector a decentralized identifier[6] whose DID Document can be resolved by counterparties at runtime through the identity services.[29] This allows participants to authenticate not only who is requesting or providing data, but also which public key and service endpoints are legitimately associated with that party. Verifiable Credentials[7] complement this identity layer by attesting ecosystem-specific roles, sector affiliations, or permissions through cryptographically verifiable claims. As a result, trust is grounded in verifiable artifacts rather than in self-declared metadata or bilateral assumptions.

A second relevant property is decentralized governance among authorized entities. Instead of delegating admission, role assignment, and issuer authorization to a single central identity provider, these functions are governed through smart contracts[5] deployed on the permissioned blockchain.[30] Authorized actors can therefore participate in governance according to transparent rules encoded at the infrastructure level. This also strengthens traceability: changes to DID records, role status, or issuer authorization are registered as governed transactions and can be audited afterwards. Consequently, the system not only supports trust establishment, but also preserves an accountable history of how trust-related decisions evolve over time, which is a core requirement for trust in data spaces.[1]

The proposed solution further enables explicit control of trusted issuers. Only issuers recognized by the governance layer are allowed to produce credentials that are accepted during verification, which reduces the risk of relying on arbitrary or unrecognized attestations. At the same time, the model preserves a clear separation between identity, credential management, and access to data. The DID layer[6] establishes who the participant is and how it can be reached; the credential layer[7] proves which attributes or permissions that participant holds; and connector-level policies determine whether access to a given dataset should be granted. This separation of concerns reduces coupling between identity management and data-sharing enforcement, while also lowering dependency on a single centralized IdP. In practical terms, this improves resilience, limits institutional lock-in, and supports a more federated trust model for data-space participation, aligned with the approaches promoted by IDSA and Gaia-X.[12,13]

Governance analysis

Encoding the TRUSTEE, STEWARD, and ENDORSER roles on-chain[30] has important governance implications for the data space. First, it means that privileged actions are not granted through opaque off-chain administrative decisions, but through rules that are explicitly enforced by smart contracts.[5] In practice, these roles define which entities are allowed to authorize issuers, endorse participants, or execute other governance-sensitive operations. By keeping these permissions on-chain, the proposed solution makes governance decisions transparent to the consortium members and reduces the possibility that one organization could unilaterally redefine trust relationships without leaving an accountable record.

This design also provides a clear mechanism for revoking trust. If a participant, steward, or issuer should no longer be recognized as trustworthy, the corresponding authorization or role can be updated or revoked through a governed blockchain transaction.[5] Once this change is committed, the affected entity immediately loses the ability to perform the privileged actions associated with that status. In the same way, if an issuer ceases to be trusted, newly presented credentials[7] from that issuer should no longer be accepted by the verification logic, even if the credential itself remains cryptographically well formed. Thus, cryptographic validity and governance-level acceptability are treated as separate conditions, which is important for maintaining operational control over the ecosystem.

Another consequence of storing governance state on-chain is that changes become auditable by design.[5] Role assignments, role removals, DID registrations, and issuer-authority updates are all represented as transactions or contract state transitions that can be inspected afterwards. This makes it possible to reconstruct who authorized whom, when a trust

Table 7. Threat analysis of the proposed solution.

Threat	Impact	Mitigation in the proposed solution	Residual risk
Malicious participant registration	Fake connector joins data space	Endorser/governance approval	Governance collusion
Compromised issuer	Invalid credentials issued	Trusted issuer registry, revocation	Detection delay
DID document tampering	Wrong endpoint/key	On-chain anchoring/signatures	Compromised keys
Proxy compromise	Incorrect DID adaptation	Verification, logging, redundancy	Availability risk
Validator collusion	Governance manipulation	Permissioned validator set/QBFT assumptions	Consortium trust required

relationship changed, and which governance decision was in effect at a given point in time. Such traceability is especially relevant in regulated environments,[8,9] where participants may need to justify why a connector or issuer was considered valid at the moment of a specific interaction.

Finally, the proposed solution prevents arbitrary identity registration by restricting DID registration[6] and related governance actions to entities that hold the required on-chain permissions.[5] In other words, not every participant can register identities or assign trust indiscriminately; only actors recognized by the governance layer are allowed to create or endorse entries that become valid within the ecosystem. This constraint helps preserve the integrity of the trust domain and avoids the emergence of unauthorized or fictitious identities. More broadly, it ensures that admission into the trust framework follows a governed procedure rather than an open, permissionless registration model.[1]

Threat analysis

Table 7 summarizes a representative set of threats affecting participant admission, issuer trust, DID integrity,[6] intermediary components, and the governance substrate of the permissioned network[30] itself. The proposed solution mitigates malicious participant registration through endorsement and governance approval, thereby preventing arbitrary connectors from joining the data space without prior authorization. Compromised issuers are constrained through the trusted issuer registry and revocation mechanisms for Verifiable Credentials,[7] although a residual window remains until the compromise is detected and the revocation state is effectively enforced by verifiers.

The table highlights that several threats are addressed by combining cryptographic guarantees with governance controls. DID document tampering is mitigated by anchoring identity state on-chain[6,5] and by relying on signatures to authenticate updates, which makes unauthorized modification substantially harder than in a mutable off-chain registry. Proxy compromise is handled through verification steps, audit logging, and the possibility of redundant deployments, although availability and correct adaptation of DID information remain exposed if a proxy instance is disrupted or behaves incorrectly.

At the governance layer, the main residual risks arise from privileged actors rather than from ordinary participants. Validator collusion or governance collusion could still distort trust decisions, which means that the proposed solution ultimately depends on the integrity assumptions of the permissioned consortium.[30] In this sense, the proposed solution does not remove trust requirements altogether; instead, it relocates them from a single opaque identity provider toward a smaller and more explicit set of governed actors whose actions are auditable in the data-space trust model.[1,12] This redistribution of trust is a key security property of the proposal, since it makes failures and abuses more visible and easier to contain through revocation and governance procedures.

Discussion

This architecture, by utilizing the advantages of blockchain and the interoperability with Verifiable Credentials model, presents a series of advantages over other initiatives.

Strengths

The architecture presented in this paper addresses a fundamental tension in federated data spaces: enabling participants to verify each other's identity and attributes without delegating that verification to a central authority. By anchoring

governance in a permissioned blockchain,[30] the system makes participation conditions — roles, issuer authorization, and identity registration — transparent and auditable to all members of the ecosystem. Unlike models based on a single trusted registry, changes to governance state are recorded immutably and are visible to every participant, providing a traceable audit trail without requiring out-of-band coordination.

The use of Verifiable Credentials[7] as the primary trust artifact enables a further separation of concerns: issuers certify attributes, connectors evaluate them, and participants retain control over what they disclose. This is consistent with the data sovereignty principles promoted by initiatives such as Gaia-X[13] and the International Data Spaces Association.[12] The integration with Eclipse Dataspace Components[25] reinforces this alignment: EDC is the most widely adopted connector framework in European data space deployments, and its native support for Verifiable Credentials and the Dataspace Protocol[26] means that the proposed trust layer can be adopted as a reusable building block without replacing the existing connector infrastructure.

Challenges and open questions

The primary practical limitation of this approach is operational complexity. Deploying the trust layer requires maintaining a permissioned blockchain network, on-chain governance contracts, a DID registry,[6] a credential issuer service, and the corresponding connector extension — each with its own operational lifecycle. This overhead is justified in ecosystems where trust decisions carry legal or compliance weight, as demanded by frameworks such as the GDPR[8] and eIDAS,[9] but it raises the barrier to adoption in simpler data-sharing scenarios.

Finally, the evaluation presented in this work was conducted in a controlled environment. Validating the architecture in a production data space — one that integrates independently deployed connectors, established identity services, and participants governed by different authorities — remains as future work. Such a setting would allow for the assessment of governance scalability, cross-domain trust portability, and the practical cost of operating the trust layer under real ecosystem conditions.

Threats to validity

The contribution of this paper combines two levels: a reference trust architecture for federated data spaces and its concrete realization through the implementation discussed in Section 6. Accordingly, threats to validity arise both from the empirical evaluation of the implemented solution and from the interpretation of the resulting evidence in relation to the broader architectural claims. The following discussion makes these limits explicit.

Construct validity

A first threat concerns whether the chosen measurements capture the intended properties of the proposal. The experimental evaluation in Sections sec:methodology and 8 focuses primarily on latency and operational overhead during onboarding and recurrent connector interactions. These metrics are appropriate for assessing the runtime cost of the trust layer, but they do not exhaust the notion of trust addressed by the paper. The contribution also concerns issuer governance, auditable role management, controlled participation, and policy-aware interaction, which are only partially expressible through latency measurements alone.

To reduce this threat, the quantitative results are complemented by the security, governance, and trust analysis in Section 9. Even so, some properties of the proposal remain analytically argued rather than empirically measured. In particular, institutional trust, governance robustness, and the effectiveness of revocation and issuer-control procedures are discussed from an architectural perspective, but not validated through long-term operational studies or adversarial field exercises.

A related construct-validity limitation is that the evaluation uses connector-level operations as proxies for the practical effect of the trust layer. This is suitable for studying operational integration, but it does not fully capture other relevant dimensions such as user-facing credential management complexity, organizational onboarding effort, or the legal and procedural burden associated with governance in real data-space ecosystems.

Internal validity

A second threat affects the causal interpretation of the measured results. The experiments were conducted in a controlled local environment in which multiple components shared the same host. While this setup improves repeatability, it also introduces possible confounding factors such as container scheduling effects, local resource contention, cache reuse, runtime warm-up behavior, and background activity on the host machine. These factors may influence absolute latency values independently of the trust architecture itself.

The measurement procedure described in Section sec:methodology mitigates this risk through repeated automated executions, warm-up phases, fixed measurement boundaries, and the use of the same environment for both the baseline and trust-enhanced scenarios. This makes the relative comparison between configurations more reliable than any single raw latency value. Nevertheless, some internal effects cannot be fully eliminated, especially for very short-lived operations whose duration may be sensitive to local runtime conditions.

Another internal-validity limitation is that the evaluation isolates the implemented trust path under controlled conditions rather than under realistic background load. As a result, the reported overheads are best interpreted as measurements of architectural behavior in a stable environment, not as exact predictions of performance under arbitrary production workloads.

External validity

The main threat to external validity concerns generalizability. The empirical results were obtained from a concrete realization of the architecture in a specific technical stack and a minimal deployment topology. In particular, the evaluation involves a single host, two connectors, one credential issuer, and a permissioned blockchain deployment with only one validator node. This is sufficient to demonstrate feasibility and to measure the local overhead of the approach, but it does not reproduce the network, organizational, and governance complexity of a production-scale federated data space.

Consequently, the observed latencies should not be interpreted as universal performance figures for all blockchain-governed SSI solutions. In larger deployments, behavior may differ due to multi-node consensus, geographically distributed participants, concurrent access patterns, heterogeneous connector versions, more complex credential schemas, or stronger governance constraints. The same limitation applies to trust portability across sectors and jurisdictions: although the architecture is designed to support governed interoperability, the current evaluation does not yet demonstrate it across multiple independent ecosystems.

A further limitation is that the implementation is integrated with Eclipse Dataspace Components. This choice is justified by the practical relevance of EDC in current data-space initiatives, but it means that some observations are necessarily influenced by the behavior, maturity, and integration model of that connector framework. The architectural principles of the proposal are intended to be broader than a single implementation stack, but this broader portability is not empirically validated here.

Conclusion validity

A final threat concerns the strength of the conclusions drawn from the available evidence. The results support claims about technical feasibility, connector-level integration, and the separation between governance-intensive and runtime-intensive costs. However, they do not amount to a definitive comparative ranking against all alternative trust and identity approaches for data spaces.

The evaluation compares the trust-enhanced deployment primarily against a baseline connector configuration rather than against a wide range of competing identity architectures. For this reason, the paper can support the claim that the proposed solution is operationally viable and that its overhead is observable and bounded in the tested environment, but not the stronger claim that it is categorically superior to all alternative decentralized or federated identity models.

Similarly, the threat analysis in Section 9 is architectural and design-oriented rather than the result of a dedicated penetration-testing campaign or formal verification effort. This does not invalidate the analysis, but it does limit the extent to which security conclusions should be read as empirical proof of resilience under hostile real-world conditions.

Taken together, these threats do not undermine the contribution of the paper, but they bound its interpretation. The work provides evidence that the proposed trust architecture can be implemented, integrated, and evaluated in a connector-based data-space setting, while also showing that broader validation in multi-domain, multi-node, and production-like environments remains a necessary next step.

Conclusion and future work

The identity solution presented on this paper covers core needs of Data Spaces, such as a decentralized architecture, independent of centralized entities; while maintaining compatibility with the main connectors existing in the ecosystem and, at the same time, with the main identity standard in the European Union, the Verifiable Credentials. The implementation of this model within Data Spaces would allow for interoperability with all architectures that use Verifiable Credentials as their standard.

The model has already been tested in controlled environments. In order to spread adoption of this solution, next steps include its testing on operative and production ready Data Spaces. This would provide useful insight on the real world needs and next main aspects to cover.

Alongside this, making a comparison with other identity solutions being developed for Data Spaces, measuring times of response or load, would allow to detect further strong and improvable points.

Ethics and consent statement

This study did not involve human participants, human data, interviews, surveys, focus groups, user testing, clinical data, or personal identifiable information. Therefore, ethical approval and informed consent were not required for this research.

Data availability

Data sharing is not applicable to this article as no datasets were created or analysed in this research.

Extended data

Extended data includes the software release of the Smart Contracts for the IDBlock platform is archived at Zenodo DOI https://doi.org/10.5281/zenodo.20792518.[41] The archived repository contains the code used in the article to compare values between Eclipse Dataspace Components (EDC) Connector as it is in their repository version 0.14, with EDC using the IDBLOCK's Smart Contracts to manage the identity of the actors.

They include README files, Smart Contracts code and test code. Manuscript tables are derived from these Smart Contracts and measuring their execution times in the conditions described.

Software availability

Reference implementation: sc-idblock. Public repository: DigitelTS/sc-idblock. Archived software release DOI: https://doi.org/10.5281/zenodo.20792518.[41] Repository license: MIT. The archived release includes source code, tests, README, LICENSE, CITATION.cff and MANIFEST.in.

Code test can be performed by cloning DigitelTS/sc-idblock, and running *npx hardhat compile* and *npx hardhat test* from the repository root.

Reproducibility

The measures presented in the manuscript are calculated by executing the EDC Connector solution, along with the Smart Contracts implementation, on deployed on a single machine equipped with an 14th-generation Intel Core i9 processor and 32 GB of RAM. All components were executed locally, with both EDC connectors running on the same host and communicating through localhost. The blockchain layer was implemented using a single Hyperledger Besu node deployed in a Docker container and configured as a private network with chain ID 1337.

With the Smart Contracts deployed on a Hyperledger Besu running locally on port 8545 execute the command *npm run deploy:local* to deploy the compiled Smart Contracts on it and test them.

The setup included only one validator node. The configuration used a block period of 2 seconds, an epoch length of 30000, and a request timeout of 4 seconds. Gas-pricing constraints were eliminated by setting the base fee to zero.

Acknowledgements

This research has been supported by the project "A catalyst for EuropeaN ClOUd Services in the era of data spaces, high-performance and edge computing (NOUS)", Grant Agreement Number 101135927. Funded by the European Union, views and opinions expressed are, however, those of the authors only and do not necessarily reflect those of the European Union. Neither the European Union nor the granting authority can be held responsible for them.

This article was made possible by the results of the IDBLOCK project (TSI-100130-2024-0143), funded by the European Union – Next Generation EU, as part of the Recovery, Transformation, and Resilience Plan, through the Sectoral Data Spaces Program of the Ministry for Digital Transformation and the Civil Service.

During the preparation of this manuscript, ChatGPT was used for formatting support, specifically transforming from a LaTeX format to one compatible with the ones accepted by the journal. All authors reviewed, edited, and approved the final manuscript and take full responsibility for its content.

References

1. Huber M, Wessel S, Brost G, *et al.*: **Building Trust in Data Spaces.** *Designing Data Spaces.* Cham: Springer; 2022.
Publisher Full Text

2. Urovi V, Jaiman V, Angerer A, *et al.*: **LUCE: A Blockchain-based data sharing platform for monitoring data license accountability and compliance.** 2022.
Publisher Full Text

3. Zhang C, Liu Y, Xu M, *et al.*: **Trans-Border Trusted Data Spaces: A General Framework Supporting Trustworthy International Data Circulation.** 2025.
Publisher Full Text

4. Menéndez R, Muñoz-Arcentales A, Salvachúa J, *et al.*: **Next Generation Authentication for Data Spaces: An Authentication Flow Based On Grant Negotiation And Authorization Protocol For Verifiable Presentations (GNAP4VP).** Shakshuki E, editor. *Procedia Computer Science.* 2025; vol. **265**: pp. 226–235.
Publisher Full Text

5. Taherdoost H: **Smart Contracts in Blockchain Technology: A Critical Review.** *Information.* 2023; **14**(2): 117.
Publisher Full Text

6. W3C: *Decentralized Identifiers (DIDs) v1.0.* W3C Recommendation; 2022. last accessed 2026/04/13.
Reference Source

7. W3C: *Verifiable Credentials Data Model v1.1.* W3C Recommendation; 2022. last accessed 2026/04/13.
Reference Source

8. **European Parliament and Council of the European Union: Regulation (EU) 2016/679 on the protection of natural persons with regard to the processing of personal data and on the free movement of such data (General Data Protection Regulation).** *Official Journal of the European Union, OJ L.* 4.5.2016; **119**: pp. 1–88. last accessed 2026/04/13.
Reference Source

9. **European Parliament and Council of the European Union: Regulation (EU) 2024/1183 amending Regulation (EU) No 910/2014 as regards establishing the European Digital Identity Framework (eIDAS 2.0).** *Official Journal of the European Union, OJ L.* 2024; **1183**: 30.4.2024. last accessed 2026/04/13.
Reference Source

10. European Commission: *Directorate-General for Communications Networks, Content and Technology: The European Digital Identity Wallet Architecture and Reference Framework (ARF), v1.4.0.* Publications Office of the European Union; 2024. last accessed 2026/04/13.
Reference Source

11. **IDBLOCK: Identidad Digital y trazabilidad descentralizada en espacios de datos.** last accessed 2026/04/13.
Reference Source

12. International Data Spaces Association: last accessed 2026/04/13.
Reference Source

13. **Gaia-X AISBL: Gaia-X Architecture Document – 22.10 Release.**2022. last accessed 2026/04/13.
Reference Source

14. **Catena-X, Your Automotive Network.** last accessed 2026/05/21.
Reference Source

15. Eona-X: **Data sharing for logistic, mobility and tourism.**
Reference Source

16. International Data Spaces Association: **Gaia-X and IDS.** last accessed 2026/05/21.
Reference Source

17. Fraunhofer-Gesellschaft: last accessed 2026/05/21.
Reference Source

18. Mobility Data Space: last accessed 2026/05/21.
Reference Source

19. Data Spaces Support Centre: last accessed 2026/04/13.
Reference Source

20. IDS Reference Architecture Model: last accessed 2026/05/21.
Reference Source

21. Verifiable Credentials Framework: **Make information easy to verify, and impossible to fake.** last accessed 2026/05/21.
Reference Source

22. **Verifiable Credential Formats in the EUDI Wallet - W3C VC DM and ISO 18013-5 mDL/mDoc.** last accessed 2026/05/21.
Reference Source

23. European Blockchain Services Infrastructure (EBSI): last accessed 2026/05/21.
Reference Source

24. Verifiable Credentials Framework: last accessed 2026/05/21.
Reference Source

25. Eclipse Foundation: **Eclipse Dataspace Components (EDC) – open-source dataspace connector framework.** 2023. last accessed 2026/04/13.
Reference Source

26. Eclipse Foundation and International Data Spaces Association: Eclipse Dataspace Protocol (DSP): 2024. last accessed 2026/04/13.
Reference Source

27. Eclipse Foundation: **EDC Documentation – Control Plane.** last accessed 2026/05/06.
Reference Source

28. Eclipse Foundation: **EDC Documentation – Control Plane Entities.** last accessed 2026/05/06.
Reference Source

29. Eclipse Foundation: **EDC Documentation – Identity Hub.** last accessed 2026/05/06.
Reference Source

30. Hyperledger Foundation: **Hyperledger Besu – Enterprise Ethereum Client.** 2023. last accessed 2026/04/13.
Reference Source

31. Hyperledger Foundation: **Besu Documentation – Configure QBFT Consensus.** last accessed 2026/05/06.
Reference Source

32. Kalibera T, Jones R: **Rigorous Benchmarking in Reasonable Time.** *Proceedings of the 2013 International Symposium on Memory Management.* ACM; 2013; pp. 63–74.
Publisher Full Text

33. Cooper D, Santesson S, Farrell S, *et al.*: **Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile. RFC 5280.** May 2008.
Reference Source

34. Schleimer AM, Jahnke N, *et al.*: **Architecture Design Options for Federated Data Spaces.** 2023.
Publisher Full Text

35. Otto B: **A federated infrastructure for European data spaces.** 2022.
Publisher Full Text

36. Mombelli D: *General-Purpose Zero-Knowledge Proofs for Verifiable Credentials.* ETH Zurich; 2025. Master's thesis.
Reference Source

37. Commission Nationale de l'Informatique et des Libertés (CNIL): **Blockchain and the GDPR: Solutions for a responsible use of the blockchain in the context of personal data.**
Reference Source

38. Jarke M, Quix C: **Federated Data Integration in Data Spaces.** Otto B, ten Hompel M, Wrobel S, editors. *Designing Data Spaces.* Cham: Springer; 2022.
Publisher Full Text

39. Blockchain And the GDPR: **A Thematic Report prepared by the European Union Blockchain Observatory and Forum.** 2018.
Reference Source

40. **IDBLOCK Governance Smart Contracts.** last accessed 2026/06/03.
Reference Source

41. Alaejos Herrero L, Ruiz Molina M, Molinero Moreno D, *et al.*: **IDBLOCK Smart Contracts. In Open Research Europe (Version 1).** *Zenodo.* 2026.
Publisher Full Text

Open Peer Review

Current Peer Review Status: ✔

Version 2

Reviewer Report 01 September 2026

https://doi.org/10.21956/openreseurope.26748.r78427

© **2026 Pincheira M.** This is an open access peer review report distributed under the terms of the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

✔ **Miguel Pincheira**
Fondazione Bruno Kessler, Trento, Italy

The manuscript proposes a blockchain-governed Self-Sovereign Identity (SSI) trust architecture for federated data spaces, integrated with Eclipse Dataspace Components (EDC).

The core contribution is a trust layer combining decentralized identifiers, Verifiable Credentials, and permissioned blockchain governance (Hyperledger Besu / QBFT), with an identity proxy mediating DID resolution between the connector runtime and the trust infrastructure.

The work is motivated by a clearly identified research gap, structured around four research questions that map cleanly onto the trust model, architecture, EDC integration, and evaluation. A reference implementation (IDBLOCK) is evaluated on latency across onboarding and runtime operations, demonstrating the expected separation between costly on-chain governance actions and lightweight runtime resolution.

Overall Assessment:

The paper is in good shape, both structurally and methodologically. The research-question-to-section mapping is clean, the problem statement earns the contribution through a well-constructed comparison of existing approaches, the architecture is well-decomposed with clearly defined trust boundaries, and the threats-to-validity section is honest about the single-node, two-connector testbed rather than overstating the results.

The following 7 concerns are minor, mostly cosmetic or consistency issues that do not affect the validity of the work. Only one point (C1) requires clarification from the authors regarding the experimental setup.

C1) Sample size inconsistency (needs clarification). The "Workloads and scenarios" text states 1000 repetitions per operation, while "Metrics and measurement procedure" states 100 observations per operation after warm-up. These should be reconciled, since the reported percentiles (p95/p99) are more robust at the larger sample size. This is a clarification, not a methodological flaw.

C2) Terminology consistency for EDC. The expansion appears as "Eclipse Dataspace Components" in most places but as "Eclipse Data Components" in the "EDC and trust extensions" subsection, which also contains "ECD" as a typo for "EDC."

C3) Several references render as "Section sec:methodology" (the raw label) instead of the resolved number, in the Introduction, the RQ mapping paragraph, and the Threats to Validity section. Maybe a problem with the web version?.

C4) The header of Table 1 uses "Oauth2/OIDC" while the body text uses "OAuth2/OIDC." (capital A).

C5) On-chain roles introduced without prior context. The TRUSTEE, STEWARD, and ENDORSER roles first appear in the Governance analysis, whereas Table 2 defines a separate conceptual role set. A brief bridge relating the two would improve consistency.

C6) Minor grammatical items. A proofreading pass should catch instances such as "a proxy is usage between ... " (" a proxy is used between ...") and "...actors might not be able of ever fulfill ..." ("..that some actors might never be able to fulfill...")

C7) DID method notation. The Introduction specifies the identity as "did:indy:besu," but the paper does not use this notation again later. A brief confirmation that the implementation uses this method would tighten the identity-layer description.

Is the work clearly and accurately presented and does it cite the current literature?
Yes

Is the study design appropriate and does the work have academic merit?
Yes

Are sufficient details of methods and analysis provided to allow replication by others?
Yes

If applicable, is the statistical analysis and its interpretation appropriate?
Yes

Are all the source data underlying the results available to ensure full reproducibility?
Yes

Are the conclusions drawn adequately supported by the results?
Yes

Competing Interests: No competing interests were disclosed.

Reviewer Expertise: Software systems, particularly decentralized and blockchain-based architectures.

I confirm that I have read this submission and believe that I have an appropriate level of expertise to confirm that it is of an acceptable scientific standard.