Revisiting Trust Management in the Data Economy: A Road Map

Claudio A. Ardagnaa and Nicola Bena, *Università degli Studi di Milano, 20133, Milan, Italy*

Nadia Bennani, *INSA Lyon, LIRIS, CNRS UMR5205, 69621, Villeurbanne, France*

Chirine Ghedira-Guegan, *University of Lyon 3, iaelyon School of Management, INSA Lyon, LIRIS, UMR5205, 69008, Lyon, France*

Nicolò Grecchi, *Università degli Studi di Milano, Milan, 20133, Italy*

Genoveva Vargas-Solar, *CNRS, UMR5205, 69622, Villeurbanne, France*

In the last two decades, multiple information and communications technology evolutions have boosted the ability to collect and analyze vast numbers of data (on the order of zettabytes). Collectively, they have paved the way for the so-called data economy, revolutionizing most sectors of our society, including health care, transportation, and grids. At the core of this revolution, distributed data-intensive applications compose services operated by multiple parties in the cloud–edge continuum; they process, manage, and exchange massive numbers of data at an unprecedented rate. However, data hold little value without adequate data protection. Traditional solutions, which aim to balance data quality and protection, are insufficient to address the peculiarities of the data economy, including trustworthy data sharing and management, composite service support, and multiparty data lifecycle. This article analyzes how trust management systems (TMSs) can regain the lead in supporting trustworthy data-intensive applications, discussing current challenges and proposing a road map for new-generation TMSs in the data economy.

The data economy is commonly defined as *an ecosystem of players collaborating to share digital data as products and services to create applications that extract value from data.*[a] The data economy transforms the design and development of applications, prioritizing data processing, management and sharing. This shift is fostering the emergence of new platforms and environments, such as *data marketplaces* and *data spaces*, where data can be pooled and shared to maximize data quality and trustworthiness, and *distributed data management systems*, which ensure guarantees to store, manage versions, and supply data for complex analytics processes. Data marketplaces[b] are (cloud) platforms where users can engage in self-service buying and selling of data, ensuring both security and high quality. Data spaces expand on this concept to define a complete environment where data infrastructures and governance frameworks are integrated to streamline data management. Within these environments, data consumers and providers collaborate to surmount existing legal and technical barriers to data sharing, thereby securely unlocking the full potential of their data.[c]

The European Union (EU) is leading the way in this revolution, with the *European Strategy for Data*[d] and

[a] European Commission, "Communication on Building a European Data Economy," https://ec.europa.eu/newsroom/dae/document.cfm?doc_id=41205.

[b] https://joinup.ec.europa.eu/collection/elise-european-location-interoperability-solutions-e-government/glossary/term/data-marketplace.

[c] https://internationaldataspaces.org/, https://digital-strategy.ec.europa.eu/en/library/staff-working-document-data-spaces.

[d] https://eur-lex.europa.eu/legal-content/EN/TXT/?uri=CELEX\%3A52020DC0066.

Digital Object Identifier 10.1109/MIC.2024.3398403
Date of publication 9 May 2024; date of current version 2 August 2024.

other initiatives, such as the *Data Governance Act*[e] and the *Common European Data Spaces*.[f] The potential economic impact is huge, enabling companies to fully capitalize on the latent value embedded within their data. For instance, the European Health Data Space is expected to generate an impact that exceeds €10 billion over 10 years;[g] additionally, the value of the data economy in the EU is expected to nearly double between 2019 and 2025.[h]

In this context, data are exiting their silos and becoming publicly available to *data-intensive applications*, enabling advanced decision making, the creation of new products, and the development of innovative business models. Data-intensive applications consist of services operated by different, possibly unknown, providers, which consume and produce data to be later shared with other applications. These services are dynamically deployed and accessed in multicloud environments as well as fog and edge devices and perform a variety of computations that all share a common requirement: the need for data. However, data pooling and sharing lose value without robust data protection measures. A considerable R&D effort has been dedicated to provide advanced data governance solutions for (balancing) data quality and protection.[4] Although significant, this effort is inadequate in today's data economy for several reasons: 1) (unknown) data providers and consumers come and go and 2) each party has its own set of potentially conflicting requirements that must be enforced on collected and exchanged data, analytics results, metadata, and primary and secondary use of data, recursively.

The data economy is reviving the trust concerns initially brought about by the emergence of the commercial Internet in the 1990s, as famously illustrated by the meme "On the Internet, nobody knows you're a dog," a cartoon penned by Peter Steiner and featured in *The New Yorker* on 5 July 1993. This meme highlighted the absence of identity-based trust in Internet transactions. Today, within the data economy, the trust issue focuses primarily on data (e.g., provenance and quality) and sources (collection, processing, maintenance and sharing conditions, and guarantees) rather than identity and services. It considers dynamic environments where the identity and privacy of individual users are not the only objectives of a trust negotiation, thus invalidating existing trust management systems (TMSs) that were initially designed to support digital transactions between unknown and untrusted parties. We reached a paradox: *despite the growing societal demand for data economy and data-intensive applications, more adequate trust management solutions are needed to uphold robust data management, processing, and sharing conditions*.

We argue that TMSs need to be extended/generalized to fully harness the potential of the data economy by 1) maximizing the value extracted from data, 2) ensuring data quality while still guaranteeing a level of data protection, and 3) enhancing the trustworthiness of multiparty applications, according to the following objectives:

- *Support for data-intensive applications*: Enabling continuous, multiparty trust evaluation and management in composite applications.
- *Restrict* to whom *data are shared*: Eliminating the unrealistic assumptions that *any* consumer can exploit data from *any* producer without limitations, and enabling multiparty data sharing based on trust requirements.
- *Control* how *data are shared*: Empowering data owners with tools to regulate how their (primary and secondary) data are processed, managed, and shared; applying different protection mechanisms (e.g., anonymization) based on negotiated trust; and overcoming the "privacy against protection" dilemma.
- Compliance *with regulations*: Ensuring adherence to the requirements mandated by laws.

This article presents a road map that paves the way to novel TMSs for data economy and data-intensive applications. It discusses current shortages, challenges, and five research directions spanning short, mid, and long terms.

DATA-INTENSIVE APPLICATIONS

A data-intensive application's primary challenges revolve around data management and processing.[16] It is implemented as a composition of services that collect, produce, analyze, and share large numbers of data. These services are typically offered by various service providers and deployed in the cloud–edge continuum according to the four architectures in Figure 1.

- *Client–service* [Figure 1(a)]: The traditional architecture in which a *client* interacts with a *service*

[e] https://eur-lex.europa.eu/legal-content/EN/TXT/?uri=CELEX\%3A32022R0868.

[f] https://digital-strategy.ec.europa.eu/en/policies/data-spaces.

[g] https://health.ec.europa.eu/system/files/2022-05/ehealth_ehds_2022ia_resume_en.pdf.

[h] European Data Market Study 2021–2023—D2.5 Second Report on Policy Conclusions: https://ec.europa.eu/newsroom/dae/redirection/document/96294.

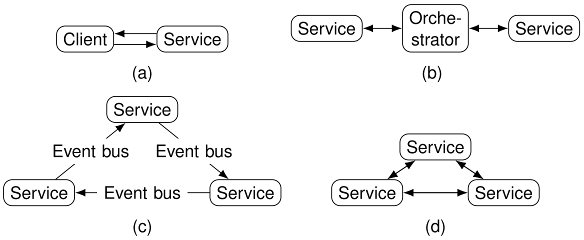

FIGURE 1. Data-intensive application architectures. (a) Client–service, (b) orchestration, (c) choreography, and (d) mesh.

(i.e., a server). The service accepts requests for retrieving data, or executes processing operations on a subset of data that it manages. This architecture typically implements access to stored data.

- *Orchestration* [Figure 1(b)]: An architecture that coordinates and manages multiple services to implement a composite service that spans various clouds. It is based on an orchestrator mediating communications between services.[17] The services manage and process data to provide results (data) that can be used as input by other services. This architecture typically implements data science workflows (e.g., for disease detection).
- *Choreography* [Figure 1(c)]: A decentralized architecture composed of multiple services that interact in a coordinated manner. The services manage and exchange data with each other using an event bus.[17] This architecture typically implements collaborative applications (e.g., across health-care providers).
- *Mesh* [Figure 1(d)]: A decentralized architecture where multiple, independent component services collaborate without any predefined logic,[15] exchanging data under specific conditions defined by the mesh. This architecture is particularly suited for challenging scenarios because of its flexibility (e.g., public health crises).

We use the terms *service* and *applications* to denote *any* computing node and its corresponding service composition (i.e., composite service), respectively. Each service can collect, prepare, analyze, and produce data within the application to enhance decision making, develop new products/processes, and create new business models. Each arrow in Figure 1 then denotes a data exchange between two services, which might initially be unknown to each other.

Reference Example

Our reference example is inspired by the European Health Data Space,[i] where hospitals and health-care facilities owned by different health-care providers across multiple countries become part of the data space to implement different medical applications built on data. For instance, hospitals analyze X-ray images for early disease diagnosis leveraging federated learning (FL);[19] health-care facilities analyze medical appointments for improved resource allocation. The retrieved results can be made available in other domains (e.g., the pharmaceutical domain) to foster data economy.

The data economy is built on an extensive exchange of data among services, forming the foundation for high-quality analytics. However, it introduces several challenges, such as data protection, ethics, and data management guarantees. In this context, the necessity to effectively exchange data and maximize utility, while addressing the specific challenges related to collected and produced data, becomes mandatory and critical for the prosperity of the data economy. Moreover, the need to establish trust between applications/services exchanging data is returning, although traditional TMSs are inadequate for this purpose.

TRUST MANAGEMENT IN DATA ECONOMY: LIMITATIONS AND MOTIVATIONS

Trust management was defined in the late 1990s by Blaze et al.[9] as *the abstract system modeling social trust in distributed environments*. Early TMSs considered client–server architectures to regulate the access of an unknown user (client) to a specific resource/service at the server.[12] More recently, TMSs have been applied to build social trust among services that support composite applications. TMSs implement a *trust negotiation protocol* to establish trust between pairs of services by matching the *trust requirements* of a service with the *service profile* of another. *Trust requirements* model the functional and nonfunctional behavior that another service must demonstrate to establish trust.[10,18] The *service profile* includes information used by a service to establish trust with another service, such as data on its functional and nonfunctional behavior (e.g., response time[1] and quality of service)[18] and reputation-based information (e.g., Azarmi et al.[8] and Kochovski et al.),[13] possibly referred to the entire application (e.g., Adewuyi et al.).[1] The *outcome* of the protocol is a binary result (success or failure)

[i] https://health.ec.europa.eu/ehealth-digital-health-and-care/european-health-data-space_en.

retrieved by matching requirements with service profiles, specifying whether the trust is established or not.[1]

Data-intensive applications compose (unknown) services that act as data providers and consumers, collaboratively sharing and processing data in long, continuously running pipelines. Data sharing occurs within data marketplaces and data spaces, or through exchange protocols that must adhere to laws; they allow owners to maintain control of their data while maximizing value extraction. Existing TMSs have numerous shortages (Ss) when applied in the data economy, as follows.

S1: *Static resource sharing*. Existing TMSs (e.g., Yu et al.)[20] statically regulate access to resources in client–service architectures. They are unsuitable for collaborative data sharing, where services dynamically interact and assume different roles (data provider/consumer), while continuously exchanging (protected) data or resources. In our reference example, the needs of different parties must be harmonized such as, for instance, interoperability of X-ray images produced using different equipment, and dynamic joining in the FL process.[19] Also, privacy guarantees at the data level must prevent patient reidentification.

S2: *Service profiles based on identity, attributes, and functional behavior*. Existing TMSs (e.g., Adewuyi et al.[1] and Cantor and Scavo)[10] are built on service profiles that contain information on static identities and attributes of users and services, focusing mainly on their functional behavior (e.g., number of successful transactions). They do not apply to the data-intensive applications whose interactions are driven by (dynamic) nonfunctional requirements (e.g., Anisetti et al.)[3] referred also to data. In our reference example, trust must be negotiated on the basis of the geographical location of applications and data, to comply with data protection regulations (e.g., the data of European citizens shall be processed in the EU or in countries for which there exists an *adequacy decision*).[j]

S3: *Static trust establishment*. Existing TMSs (e.g., Li et al.)[14] focus on static trust, which is negotiated before any communication among services occurs and is assumed to be valid for the entire application's execution and beyond. However, data-intensive applications undergo frequent changes, rendering the negotiated trust quickly outdated. Early solutions to this issue are only partial (e.g., Azarmi et al.[8] and Kochovski et al.).[13] In our reference example, new parties can join or their service profiles change over time (e.g., due to a service update).

S4: *Centralized model*. Existing TMSs (e.g., Alshehri and Hussain)[2] are centralized, with a controller managing the entire trust protocol. Centralization introduces the need to select and protect the controller, and only applies to architectures where a central component can be identified (e.g., orchestration). In our reference example, the early disease diagnosis application uses FL to guarantee a level of privacy,[11,19] while the appointment application requires a decentralized pipeline to share data and resources.

These limitations show that existing TMSs, including those recently specified,[1] inadequately address the peculiarities of data-intensive scenarios and are therefore inapplicable.

CHALLENGES IN TMSs

A set of challenges (Cs) arises in defining a TMS that addresses shortages S1–S4. We analyze these challenges according to four aspects: 1) *design and architecture*, 2) *service profile*, 3) *trust negotiation protocol*, and 4) *trust lifecycle*.

Design and Architecture

Challenges in design and architecture focus on the data produced and consumed by applications.

C1.1: *Multiparty data sharing*. In data-intensive applications, multiple parties collaboratively produce and consume data, overcoming the traditional client–server model. Data sharing is also hierarchical, where some parties collect and recursively share data related to or on behalf of other parties. TMSs should depart from server-side requirements and access control, enabling the definition of requirements at different granularity (e.g., an entire application, individual data providers, and the role of consumer/provider) (S1 and S2).

C1.2: *Nonbinary outcome*. In traditional TMSs, the outcome of a trust negotiation protocol is binary, meaning that a service is either invoked or not. This is not suitable in

[j] General Data Protection Regulation, Article 45.

data-intensive applications, where data exchange *must* be maximized to meet data-intensive applications objectives. TMSs should enable selective data sharing according to the negotiated trust level (S1). In the appointment application in our reference scenario, services necessitate data that pertain to similar geographical areas (e.g., similar population density). The application task should proceed even in the absence of a perfect match, rather than not executing the task at all.

C1.3: *Decentralized environment*. Following C1.1 and C1.2, TMSs should support decentralized requirements. The corresponding outcome should be retrieved according to the role of each service in the application (which may change over time), rather than being built on only a client–service architecture (S4).

Service Profile

A service profile includes the information used to establish trust. Data-intensive applications introduce challenges in service profile life cycle management.

C2.1: *Trustworthiness*. Existing TMSs use different solutions (e.g., blockchain)[13] to ensure integrity and authenticity of service profiles. The shift to data-intensive applications requires modeling, retrieving, managing, and distributing trustworthy information in service profiles among multiple parties, encompassing aspects beyond integrity and authenticity (S2 and S3).

C2.2: *Secure management*. Service profiles may contain sensitive information that needs to be protected without conflicting with their use in trust negotiation (S1 and S4). In our reference example, a service profile may reveal information that can be exploited by attackers.[6]

C2.3: *Derivation*. Trust establishment should build on information in service profiles as well as *secondary* information derived from the services and their data (e.g., the location of a data processing can be derived from the location of its services). Following C2.1 and C2.2, secondary information should be trustworthy and managed according to the corresponding primary information (S1, S2, and S3).

Trust Negotiation Protocol

Challenges in the trust negotiation protocol focus on how trust is concretely negotiated in data-intensive applications.

C3.1: *Flexible execution*. Traditional trust negotiation protocols are point to point or rely on a centralized source of information, which conflicts with the peculiarities of data-intensive applications. A proper protocol should be 1) capable to establish trust with and without an orchestrator, 2) resilient to changes in the architecture, and 3) efficient, minimizing the time spent on trust establishment (S4).

C3.2: *Best-effort solution*. Following C1.2, a trust negotiation protocol should maximize the chances of a positive outcome and data exchange, even if requirements are partially violated (S1). Violation costs should be affordable and evaluated case by case.

C3.3: *Conflict resolution*. Despite C1.1 and C3.2, there are situations, called *conflicts*, where requirements do not match at all (e.g., two services with *opposite* requirements). The trust negotiation protocol should properly manage conflicts (S1 and S3). In our reference example, a service in the appointment application requires preprocessed data, while another service accepts only raw data. The two services cannot jointly participate in the application unless the trust negotiation protocol solves the conflict.

Trust Lifecycle

Challenges in the trust life cycle focus on how trust is managed in data-intensive applications. Services enter and exit the corresponding applications, while their service profile can change, introducing new issues.

C4.1: *Reliability*. A change in a service can imply changes in its profile. Established trust can become spurious and must be rectified. In this process, TMSs should avoid disruptions and prevent harm to the application execution (S3 and S4). In our reference example, a service in the appointment application is moved to another provider, changing the location in its service profile and impacting the previously negotiated trust.

C4.2: *History-based trust*. Data-intensive applications implement long-running pipelines with services dynamically recruited at various stages. Requirements should be expressed

TABLE 1. Road map.

Research direction	Challenge	Shortage	Timeline	Related techniques
RD1: Requirements	C1.1: Multiparty data sharing	S1 and S2	Short, medium, and long	Role-based access control, quality of service, identity management, authorization protocols, service selection and composition, SLAs, reputation, and certification
	C2.1: Trustworthiness	S2 and S3	Short	
	C2.3: Derivation	S1, S2, and S3	Medium	
RD2: Protocol	C1.2: Nonbinary outcome	S1	Short	Partial and incremental trust, credentials management, SLAs, data governance and privacy, and risk-adaptive access control
	C3.1: Flexible execution	S4	Short	
	C3.2: Best-effort solution	S1	Short	
RD3: History	C1.3: Decentralized environment	S4	Medium and long	Blockchain, Bayesian inference, version control, and behavior prediction
	C4.2: History-based trust	S3	Medium and long	
	C4.3: Management of historical interactions	S3 and S4	Medium and long	
RD4: Conflicts	C3.2: Best-effort solution	S1	Short	MCDA, policy reconciliation, conflict resolution, version control, and consensus
	C3.3: Conflict resolution	S1 and S3	Medium and long	
	C4.1: Reliability	S3 and S4	Medium and long	
RD5: Assurance	C2.2: Secure management	S1 and S4	Short, medium, and long	Credential management, selective release, certification, and risk management
	C2.3: Derivation	S1, S2, and S3	Medium	

RD: research direction; SLAs: service-level agreements; MCDA: multicriteria decision analysis.

along all stages, taking into account the complete history of the application (e.g., recruited services and their role in the application) (S3).

C4.3: *Management of historical interactions*. As a consequence of C2.2 and C4.2, TMSs should consider past interactions observed during the application execution to evaluate the overall trust. TMSs should store 1) profiles of services currently part of the application and 2) profiles of services that left it. This challenge further increases the complexity of the entire trust negotiation protocol in C3.1–C3.3 (S3 and S4).

RESEARCH ROAD MAP

According to the identified challenges, TMSs must be revisited across three milestones composed of five research directions (RDs) RD1–RD5, as depicted in Table 1.[k]

The *first milestone* aims to address the most pressing challenges, focusing on client–service and orchestration architectures (RD1, RD2, RD4, and RD5). It should be completed in the short term (≤two years). The *second milestone* aims to initially address all challenges, with additional focus on choreography architectures (RD1, RD3, RD4, and RD5). It should be completed in the medium term (>two years, ≤five years). The *third milestone* aims to fully address all challenges, with an additional focus on mesh architectures (RD1, RD3, RD4, and RD5). It should be completed in the long term (>five years).

RD1: Requirements

Requirements must be revisited to cope with multiparty collaborations. First, every service in the data-intensive application should express requirements at different granularity levels (C1.1 and C2.1), for instance, an individual service (data provider/consumer), a group of services, and an application. Second, requirements should also be expressed on (certified) secondary information (C2.3). Existing access control methodologies (e.g., attribute-based and role-based), service-level

[k] Extended version available at https://github.com/SESARLab/revisiting-trust-management-in-the-data-economy-roadmap/blob/master/main.pdf.

agreements and reputation techniques, and certification schemes can be used for this purpose.

RD2: Trust Protocol

The trust protocol must be scalable and dynamic, occasionally prioritizing faster convergence over absolute accuracy (C3.1). On the one hand, the negotiated trust should accommodate the majority of services, which makes a perfect match rarely feasible and requires heuristic protocols (C3.2). On the other hand, services must have the ultimate choice of what to do according to the satisfaction of their requirements (C1.2).

RD3: History

The history of the data-intensive application must be considered. Requirements should be expressed on the present and past versions of the application, in terms of, for instance, recruited services and processed data (C4.2). The application history must be reliably tracked over time (C1.3 and C4.3), for example, using blockchain. When not available, the application history can be simulated using behavior prediction.

RD4: Conflicts

Conflicts can arise when two or more requirements contradict each other or services/data change during the application lifecycle (C3.3). Conflicts can be tracked using version control and reconciled using policy reconciliation, conflict resolution, and consensus. Building on RD3, TMSs should detect and manage conflicts, prioritizing 1) the *preservation* of already-established trust relationships through consensus mechanisms (C4.1) and 2) the satisfaction of the requirements of the majority of services (C3.2) without wasting computing resources, according to multicriteria decision analysis (MCDA).

RD5: Assurance

Service profiles should rely on up-to-date information retrieved according to *assurance* techniques.[5] The integration with such techniques (e.g., certification)[3] must consider how to 1) model (historical) service profiles, 2) compose and derive information from service profiles, and 3) safely manage service profiles (C2.2 and C2.3). The latter ensures that service profiles are stored and adequately protected for future negotiations, for instance, building on *selective release*[7] and *certificate composition*.[6]

To conclude, it becomes apparent that dynamicity poses the primary barrier to trust management in the data economy. The continuous changes and the management of historical service profiles and requirements point to a broader concept: *trust must be defined as a function of time, encompassing a lengthy, intricate, and continuously adapted chain of trust between services and their providers/users*. We postulate that the optimal establishment of such trust is a computationally intractable problem, as was in the case of traditional client–server architectures.[7]

ACKNOWLEDGMENTS

Research supported in part by 1) project BA-PHERD, funded by the European Union – NextGenerationEU (NGEU), under the National Recovery and Resilience Plan (NRRP) Mission 4 Component 2 Investment Line 1.1: “Fondo Bando PRIN 2022” (CUP G53D23002910006); 2) project MUSA, funded by the European Union - NextGenerationEU, under the National Recovery and Resilience Plan (NRRP) Mission 4 Component 2 Investment Line 1.5: Strengthening of research structures and creation of R&D “innovation ecosystems.”, set up of “territorial leaders in R&D” (CUP G43C22001370007, Code ECS00000037); 3) project SERICS (PE00000014) under the NRRP MUR program funded by the EU - NGEU; 4) University of Milano under the program “Piano di Sostegno alla Ricerca”; 5) the SUMMIT program Pack Ambition funded by the Auvergne Rhône Alpes region; and 6) FRIENDLY of the LIRIS lab. Views and opinions expressed are however those of the authors only and do not necessarily reflect those of the European Union or the Italian MUR can be held responsible for them.

REFERENCES

1. A. A. Adewuyi, H. Cheng, Q. Shi, J. Cao, X. Wang, and B. Zhou, “SC-TRUST: A dynamic model for trustworthy service composition in the internet of things,” *IEEE Internet Things J.*, vol. 9, no. 5, pp. 3298–3312, Mar. 2022, doi: 10.1109/JIOT.2021.3097980.
2. M. D. Alshehri and F. K. Hussain, “A centralized trust management mechanism for the internet of things (CTM-IoT),” in *Proc. Int. Conf. Broadband Wireless Comput. Commun. Appl.*, Barcelona, Spain, 2018, pp. 533–543.
3. M. Anisetti, C. A. Ardagna, and N. Bena, “Multi-dimensional certification of modern distributed systems,” *IEEE Trans. Services Comput.*, vol. 16, no. 3, pp. 1999–2012, May/Jun. 2023, doi: 10.1109/TSC.2022.3195071.
4. M. Anisetti, C. A. Ardagna, C. Braghin, E. Damiani, A. Polimeno, and A. Balestrucci, “Dynamic and scalable enforcement of access control policies for big data,” in *Proc. 13th Int. Conf. Manage. Digit. EcoSyst.*, 2021, pp. 71–78, doi: 10.1145/3444757.3485107.

5. C. A. Ardagna, R. Asal, E. Damiani, and Q. H. Vu, "From security to assurance in the cloud: A survey." *ACM Comput. Surv.*, vol. 48, no. 1, pp. 1–50, 2015.
6. C. A. Ardagna and N. Bena, "Non-functional certification of modern distributed systems: A research manifesto," in *Proc. IEEE Int. Conf. Softw. Services Eng. (SSE)*, Chicago, IL, USA, 2023, pp. 71–79, doi: 10.1109/SSE60056.2023.00020.
7. C. A. Ardagna, S. De Capitani di Vimercati, S. Foresti, S. Paraboschi, and P. Samarati, "Supporting privacy preferences in credential-based interactions," in *Proc. 9th Annu. ACM Workshop Privacy Electron. Soc.*, Chicago, IL, USA, 2010, pp. 83–92, doi: 10.1145/1866919.1866931.
8. M. Azarmi et al., "An end-to-end security auditing approach for service oriented architectures." in *IEEE 31st Symp. Reliable Distrib. Syst.*, Irvine, CA, USA, 2012, pp. 279–284, doi: 10.1109/SRDS.2012.5.
9. M. Blaze, J. Feigenbaum, and J. Lacy, "Decentralized trust management," in *Proc. IEEE Symp. Secur. Privacy*, Oakland, CA, USA, May 1996, pp. 164–173, doi: 10.1109/SECPRI.1996.502679.
10. S. Cantor and T. Scavo, "Shibboleth architecture," *Protocols Profiles*, vol. 10, no. 16, pp. 1–19, 2005.
11. F. Fu, H. Xue, Y. Cheng, Y. Tao, and B. Cui, "BlindFL: Vertical federated machine learning without peeking into your data," in *Proc. Int. Conf. Manage. Data*, Philadelphia, PA, USA, Jun. 2022, pp. 1316–1330.
12. V. E. Jones, N. Ching, and M. Winslett, "Credentials for privacy and interoperation," in *Proc. New Security Paradigms Workshop*, La Jolla, CA, USA, Aug. 1995, pp. 92–100, doi: 10.1109/NSPW.1995.492348.
13. P. Kochovski, S. Gec, V. Stankovski, M. Bajec, and P. D. Drobintsev, "Trust management in a blockchain based fog computing platform with trustless smart oracles," *Future Gener. Comput. Syst.*, vol. 101, pp. 747–759, Dec. 2019.
14. N. Li, J. C. Mitchell, and W. H. Winsborough, "Design of a role-based trust-management framework," in *Proc. IEEE Symp. Secur. Privacy*, Berkeley, CA, USA, 2002, pp. 114–130, doi: 10.1109/SECPRI.2002.1004366.
15. W. Li, Y. Lemieux, J. Gao, Z. Zhao, and Y. Han, "Service mesh: Challenges, state of the art, and future research opportunities," in *Proc. IEEE Int. Conf. Service-Oriented Syst. Eng. (SOSE)*, San Francisco, CA, USA, 2019, pp. 122–1225, doi: 10.1109/SOSE.2019.00026.
16. A. Margara, G. Cugola, N. Felicioni, and S. Cilloni, "A model and survey of distributed data-intensive systems," *ACM Comput. Surveys*, vol. 56, no. 1, pp. 1–69, 2023.
17. C. Peltz, "Web services orchestration and choreography," *Computer*, vol. 36, no. 10, pp. 46–52, Oct. 2003, doi: 10.1109/MC.2003.1236471.
18. S. Romdhani, G. Vargas-Solar, N. Bennani, and C. Ghedira, "QoS-based trust evaluation for data services as a black box," in *Proc. IEEE Int. Conf. Web Services (ICWS)*, Chicago, IL, USA, 2021, pp. 476–481, doi: 10.1109/ICWS53863.2021.00067.
19. Q. Yang, Y. Liu, T. Chen, and Y. Tong, "Federated machine learning: Concept and applications," *ACM Trans. Intell. Syst. Technol.*, vol. 10, no. 2, pp. 1–19, 2019.
20. T. Yu, M. Winslett, and K. E. Seamons, "Supporting structured credentials and sensitive policies through interoperable strategies for automated trust negotiation," *ACM Trans. Inf. Syst. Security*, vol. 6, no. 1, pp. 1–42, 2003.

CLAUDIO A. ARDAGNA is a full professor at the Università degli Studi di Milano, 20133, Milan, Italy, and director of the CINI National Lab on Data Science. His research interests include cloud–edge security, assurance, and data science. Ardagna received his Ph.D. degree in computer science from the University of Milan. He is a Senior Member of IEEE. Contact him at claudio.ardagna@unimi.it.

NICOLA BENA is a postdoctoral researcher at the Università degli Studi di Milano, 20133, Milan, Italy. His research interests are in the area of security of modern distributed systems assurance. Bena received his Ph.D. degree in computer science from the University of Milan. He is a Graduate Student Member of IEEE. Contact him at nicola.bena@unimi.it.

NADIA BENNANI is an associate professor at the INSA Lyon, LIRIS, CNRS UMR5205, 69621, Villeurbanne, France. She is a member of the DRIM group at the Laboratory of InfoRmatics in Image and Information Systems. Her research interests include privacy and trust in distributed systems and secure data sharing. Bennani received her Ph.D. degree in computer science from the University of Lille-France. Contact her at nadia.bennani@insa-lyon.fr.

CHIRINE GHEDIRA-GUEGAN is a full professor of computer science at University Lyon 3, iaelyon School of Management, INSA Lyon, LIRIS, UMR5205, 69008, Lyon, France, and deputy head of the SOC research team at the Laboratory of InfoRmatics in Image and Information Systems, National Center for Scientific Research, Lyon, France. Her research interests include large-scale integration of heterogeneous data and services, with particular attention to source servitization, privacy, trust and security, and AI in the co-evolution of distributed systems. Ghedira-Guegan received her research habilitation degree in computer science from University Claude Bernard Lyon 1. Contact her at chirine.ghedira-guegan@liris.cnrs.fr.

NICOLÒ GRECCHI is a B.Sc. student in systems and network security at the University of Milan, Milan, 20133, Italy. His research interests are in the area of trust management. Contact him at nicolo.grecchi@studenti.unimi.it

GENOVEVA VARGAS-SOLAR is a senior scientist at the CNRS and a member of the Database group at the Laboratory of InfoRmatics in Image and Information System (LIRIS) UMR5205, 69622, Villeurbanne, France. Her research interests include service-based data science management systems, data science workflows enactment guided by Service Level Objectives (SLO), query optimization on high performance architectures, and disaggregated data centres. Vargas-Solar received her Ph.D. degree in computer science from the University Joseph Fourier. Contact her at genoveva.vargas-solar@cnrs.fr

Open Access funding provided by 'Università degli Studi di Milano' within the CRUI CARE Agreement