Zubair *et al.*
EURASIP Journal on Information Security *(2024) 2024:30*
https://doi.org/10.1186/s13635-024-00178-z

EURASIP Journal on
Information Security

RESEARCH **Open Access**

Access control for trusted data sharing

Maria Zubair[1*], Maryam Sabzevari[2], Vikramajeet Khatri[2], Sasu Tarkoma[1] and Kimmo Hätönen[2]

Abstract

In the envisioned 6G landscape, data sharing is expected to become increasingly prevalent, giving rise to digital marketplaces that foster cooperation among organizations for collecting, sharing, and processing data for analysis. These marketplaces serve as connectors between data producers and consumers, empowering multi-tenancy scenarios for seamless and secure data sharing both within and outside organizations. Given that 6G networks promise ultra-low latency, enhanced connectivity, and massive data throughput, the need for robust data access control mechanisms becomes imperative. These mechanisms ensure security and trust among entities, particularly in multi-tenant environments where multiple organizations share infrastructure and data resources. In this paper, we have designed and implemented a novel access control mechanism tailored for a distributed data streaming system developed by Nokia Bell Labs. Our approach leverages fine-grained policies, dynamic enforcement, and transparency mechanisms to enhance trust between data owners and consumers. By facilitating secure multi-tenancy data sharing, our solution contributes to the seamless exchange of data across diverse entities within the next-generation communication ecosystem. We demonstrate that our proposed access control mechanism incurs minimal overhead while ensuring data confidentiality and integrity. The introduction of such advancements in data sharing markets strengthens the overall ecosystem by providing heightened transparency and enhanced control over data, promoting collaboration and innovation in the 6G era.

Keywords Access control, 6G, Multi-tenancy, Data ownership, Data sharing, Trust

1 Introduction

Modern organizations highly value the ever-increasing data for customer behavior analysis and market demand forecasting, making it their most valuable asset [1–5]. They heavily invest in data collection and analysis for decision-making and customer reporting, yet many still lack the resources to fully leverage their data's potential [6]. Therefore, organizations must consider and implement well-designed solutions for representing and suitably exposing data. This will allow consumers, both within and outside the organization, to accurately retrieve and interpret such data.

With 6G advancements, real-time applications and services will thrive with high-speed connectivity and low latency, particularly for data collection and analysis from the Internet of Things (IoT) devices and Business Support Systems (BSS). Data sharing facilitates data exchange [7, 8] without the need for individual data collection setups, promoting multi-tenancy in the network. Multi-tenancy emerges as a significant concept in the 6G ecosystem, which allows multiple entities to share a common infrastructure while keeping their data and operations separate, promoting cost-effectiveness and resource efficiency.

Distributed data streaming systems collect and share data from various resources among organizations, ensuring effective data dissemination to interested customers. Publish/subscribe (pub/sub) paradigm [9–11] provides a suitable data-centric communication infrastructure for large-scale distributed applications, allowing subscribers to express interest and receive timely notifications from publishers, making it highly suitable for IoT applications. The pub/sub model's strength lies in its three decoupling

*Correspondence:
Maria Zubair
maria.zubair@helsinki.fi
[1] Department of Computer Science, University of Helsinki, Helsinki, Finland
[2] Nokia Bell Labs, Espoo, Finland

dimensions of time, space, and subject [9, 10], increasing system scalability by eliminating direct communication dependencies between data-sharing parties. This paper discusses a Service-Oriented Architecture (SOA) [12], utilizing pub/sub mechanisms with microservices for improved scalability [13, 14].

Distributed data streaming systems, in addition to data sharing, manage data access among organizations, but legacy practices are inadequate for the growing volume and complexity of data [7]. As we move towards the 6G era, technological advancements promise ultra-low latency, enhanced connectivity, and massive data throughput. This new landscape will need to support a vast number of connected devices and integrate cutting-edge innovations like artificial intelligence (AI), machine learning (ML), and the Internet of Things (IoT) [15]. Managing this environment will require sophisticated data management strategies, especially in multi-tenant scenarios where multiple organizations share infrastructure and data, which can be utilized by various entities [16].

Moreover, the dynamic reconfigurability inherent in 6G networks, driven by AI and ML-based decision-making, will demand robust access control measures to ensure security and trust [15]. This necessitates the development of novel access control mechanisms specifically tailored for distributed data streaming systems. These mechanisms must incorporate fine-grained policies, dynamic enforcement, and transparency features to promote trust between data owners and consumers [17]. Such advanced access control systems are essential for secure and efficient data exchange, addressing the unique challenges posed by the 6G ecosystem. This not only protects sensitive information but also enhances the overall reliability and functionality of the network, enabling seamless and secure interactions among the myriad of devices and systems operating within this next-generation technological framework.

In today's data-driven landscape, organizations utilize BSS and IoT data for business opportunities, informed decisions, and valuable insights [5, 7, 18]. Trust in data sharing becomes crucial for secure collaboration, enabling multi-tenancy and multi-player networks that foster innovation and data-driven decision-making. Several research efforts focus on designing data markets to facilitate data sharing [7]. Trust refers to the establishment of reliance and confidence through a robust access control mechanism. It ensures secure collaboration, multi-tenancy data sharing, and transparent practices, fostering innovation and data-driven decisions in a multi-player network such as 6G.

To this end, a well-defined access control mechanism to empower the data owners and creating trust is essential. To summarize, our work offers the following contributions:

- Providing a comprehensive review of data sharing systems and access control requirements.
- Presenting the proposed access control system, which extends the state-of-the-art access control models, including XACML and NGAC, with a strong focus on data ownership and transparency.
- Conducting experiments to evaluate the performance of the proposed access control system.

The access control requirements facilitate the recognition of the needs of stakeholders and the systems used for sharing data. The insights gained from this research shed light on the design of reliable and trustworthy data sharing systems, offering valuable guidance for future studies in this field. The rest of the paper is organized as follows: Section 2 provides a discussion of related work on access control mechanisms. In Section 3, we present the proposed solution and its evaluation. Section 4 brings the limitations of current work and future directions. Finally, conclusion is presented in Section 5.

2 Background and related work

International Data Space Association (IDSA) develops standards for data exchange [8]. They introduced the International Data Space Reference Architecture Model (IDS-RAM), promoting trustworthy data-driven ecosystems, products, and services based on European principles. IDS-RAM includes the International Data Space (IDS) Connector, ensuring data sovereignty, with data providers defining access control and usage policies. Other works focusing on developing data markets include Open Data Initiative (ODI) [19], DIGITEUROPE [20], and Digital Universe [21]. Current data-sharing systems face a trust gap between big data collectors and providers [22–24] due to lack of transparency and accountability. Efforts to bridge this gap include empowering data providers and introducing regulations like GDPR [25], which allows data collection with consent and the right to withdraw consent. In addition, enhancing transparency and accountability in data sharing systems is crucial [26], as awareness and control play key roles in building trust among stakeholders [27].

2.1 Access control

Access control mechanisms ensure compliance with predefined resource access policies, encompassing identification, authentication, authorization, and access decision [28].

Among access control models, Mandatory Access Control (MAC) [29] stands out as a static approach that assigns clearance labels to data resources and users based on authorized actions. Users have at least one clearance level determining their data access. However, MAC restricts data owners from granting individual access rights. In contrast, Discretionary Access Control (DAC) [30] provides more owner control using Access Control Lists (ACLs) to define precise permissions for users. For example, NTFS allows file owners to specify a subject group with specific resource permissions. Role-Based Access Control (RBAC) [31] assigns roles to users within an organization, simplifying access management based on job titles, security levels, or departments. In contrast, Attribute-Based Access Control (ABAC) [32] enables dynamic and context-aware control, using resource, subject, and environmental characteristics to determine data access. Usage Control (UCON) [33] provides fine-grained real-time access control, utilizing attributes related to data usage. Both ABAC and UCON define access rules based on attributes of resources and subjects, with UCON supporting additional attributes for data usage. This continuous access control checks user-specified constraints for decision-making.

2.2 Trusted data sharing

In the era of advanced technologies like 6G, data sharing is vital for success in diverse ecosystems such as smart cities. These environments heavily rely on data from sensors and devices to generate valuable insights for decision-making [26]. However, the data shared within such multi-player environments may contain sensitive information, necessitating robust privacy and security measures [34].

Privacy is closely linked to data owners' control over their data, encompassing access control, trust, and transparency in provenance. Estivill et al. [35] explored techniques empowering social network users with data privacy awareness and control. They distinguished between confidential and non-confidential attributes, enabling users to decide sensitive information. Strict access control is enforced for confidential attributes based on user preferences. Additionally, empowering data owners involves transparency in data provenance [36–38], capturing data origin, lineage, and influencing entities. This transparency aids data owners in understanding changes and entities involved, facilitating data-sharing decisions.

IDSA's reference architecture [8] employs data owner-defined ABAC policies, including connector identity, attributes, and security profiles, and emphasizes data usage control for processing obligations. Carminati et al.'s privacy reference model [39] enhances user control in IoT platforms through hierarchical data categories and purposes organized in tree structures. Users can define fine-grained privacy preferences at attribute levels, specifying data access control and joint access constraints to prevent information leakage. Automated derivation of privacy preferences simplifies policy setup for derived data, making it easier for users to manage their data privacy.

Several research works have proposed enforcement mechanisms and access control models to address the requirements of big data applications, including IoT and data streams. Colombo and Ferrari introduced an enforcement mechanism utilizing enforcement monitors for runtime access control enforcement on MQTT-based IoT ecosystems [40, 41]. Carminati et al. proposed an access control model for data streams that employs a query rewriting approach. Users can submit queries to read or aggregate data, and these queries are evaluated by a Query Rewriter component for access permissions. Based on the evaluation result, the query is executed either partially or completely, ensuring that only authorized data is included in the query results. The evaluation process also considers temporal access constraints, such as permitted time windows for access control [42].

Guerriero et al. [43] proposed a framework for automatic code rewriting that uses a query to enforce privacy policies in data-intensive applications. Their focus area was access control on accessing the sensitive data. However, there is no mechanism defined for access control on data transformations in the system. Studies have shown that access control models can be extended to provide additional features, such as detection of the anomalous behavior of users [44, 45] and defining trust domains by grouping together nodes which share the same privacy preservation expectations and are trusted not to leak private information outside the trusted domain [46].

GAIA-X is a federated secure data infrastructure aimed at providing a networked data infrastructure which can meet the most demanding digital sovereignty criteria while remaining future-proof. It provides the standard requirements for data exchange and emphasizes the need of monitoring and logging capabilities in the data exchange service to ensure the sovereignty of data [47, 48].

Even though other approaches have implemented various access control mechanisms for data sharing in different environments, they often lack a comprehensive solution tailored specifically for multi-tenant networks like 6G. In scenarios where there are multiple entities or tenants coexisting within the same network, traditional access control methods may fall short in meeting the unique requirements and complexities of such a setup. This paper addresses this limitation by proposing an access control model explicitly designed for multi-tenant environments, offering distinct advantages over existing approaches. In particular, our proposed access control model enables data processing at different stages and by various subjects, each governed by distinct access control rules. This allows multiple organizations to collaborate, share, and process data within a unified access control mechanism. Furthermore, the use of standard policy components and simplified access policies ensures efficient policy management and enforcement, vital for a network with numerous tenants and data streams. Finally, the model's optimized decision-making and enforcement processes minimize latency and resource overhead, enabling scalable and efficient data sharing among tenants.

3 Proposed solution

In this section, we present the key requirements of access control, propose our solution based on these requirements, and evaluate our proposed solution.

3.1 Key requirements of access control solutions for IoT and big data

Notably, for IoT and BSS data sharing scenarios, there are specific criteria that access control mechanisms must address [40, 49–52].

3.1.1 Policy specification (choice of access control model)

One crucial aspect is the need for defining fine-grained policies that can handle dynamic attributes in heterogeneous environments. This necessitates context-based access control, which considers environmental conditions, such as time periods and geographical locations [50]. ABAC is well-suited to fulfill this requirement, as it allows access rights to be defined based on dynamic attributes at a granular level [52]. An established oasis standard for ABAC is eXtensible Access Control Markup Language (XACML), which offers a well-structured policy-based reference architecture [53], distinguishing enforcement, decision-making, and policy management. This architecture is widely adopted for its support of ABAC and its facilitation of independent component development. Next Generation Access Control (NGAC) [54] is another standard, focusing on ABAC mechanism which employs data relations and attributes for fine-grained access control. It aims for real-time adaptability and enhanced expressiveness through operations, functions, and constraints within the policy language. It particularly focuses on dynamically evolving scenarios where access decisions require immediate context-aware adjustments.

3.1.2 User-centric mechanism

Furthermore, to cater to data sharing requirements in IoT and BSS scenarios, the system should prioritize a user-centric approach. This allows all stakeholders, even those with limited security expertise, to effectively handle access control rules. Partial or automatically generated policies can aid in achieving this objective [55, 56].

3.1.3 Efficiency

Moreover, policy management should be streamlined, with a centralized administration point and minimal policy definitions [57]. To ensure efficiency in big data processing systems, access control mechanisms should be designed to avoid becoming bottlenecks for overall system performance. Efficiency can be measured by considering factors like decision and enforcement capabilities, time taken for decisions, and communication overhead among access control components [58]. Techniques such as view-based access and query rewriting have been explored to reduce latency overhead, but further work is needed to optimize efficiency, given the complexity of real-world datasets [56, 57, 59].

3.1.4 Reliability and interoperability

Finally, reliability and interoperability are critical aspects of the access control system. Regardless of the deployment method, reliability and availability of components involved in policy evaluation and enforcement need to be ensured. If any access control node fails, the system should still be operational and provide the necessary functionality for ongoing data streams within the system. However, for any change in ongoing data stream or new subscription for a data stream, it needs an access control node to validate change requests.

3.2 System overview

We put forward an extension of the access control within the Distributed Data Streaming System (DDSS) developed by researchers at Nokia Bell Labs [60, 61]. The DDSS system follows a publish/subscribe paradigm which facilitates efficient communication and data transmission between producers and consumers of data. In Fig. 1, we illustrate a schematic diagram of the data processing components within a data streaming system. As the system can scale with multiple number of these components, the processing is distributed to avoid the bottleneck caused by a centralized system.

A data source can be any sensor, network element, or user device. The data produced by all data sources is sent to a Data Fetcher (DF). DF can fetch and receive data from internal and external sources. Internal sources can be data sources from internal network elements, while external sources can be, for example, any IoT data source using network services. It also applies processing functions on raw data before forwarding it to a Data Switch (DS) comprising of pluggable publish/subscribe brokers, such as Apache Kafka or ActiveMQ. It receives pre-processed data from DFs and routes it to Data Hubs (DHs)

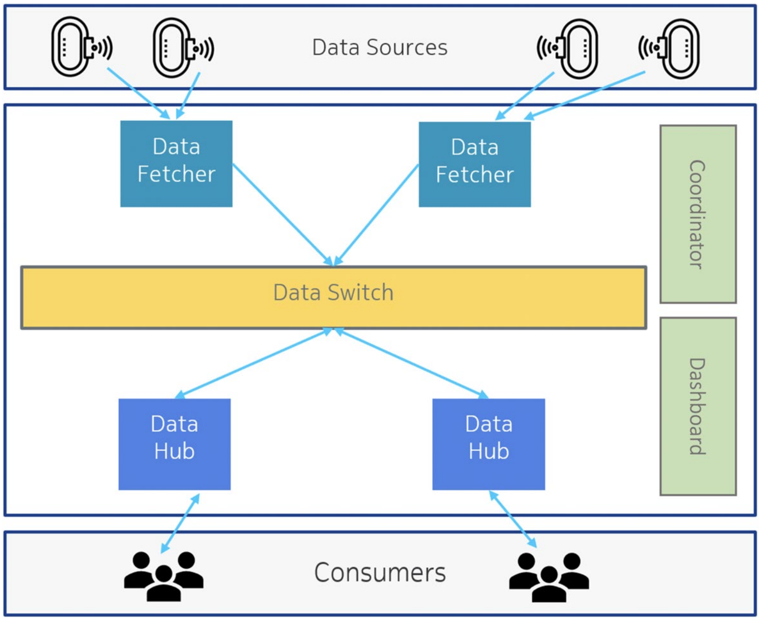

Fig. 1 Architecture

based on user subscriptions. A DH is a gateway for end-users where users can request for data subscriptions.

Upon successfully creating a subscription, DH receives data and forwards it to end-users. There are two management components in the system, namely Coordinator and Dashboard. The Coordinator is responsible for managing and coordinating activities of all other components. Its functions include managing data subscriptions between DF and DH. The Dashboard provides an interface for monitoring the state of the system.

Nokia Bell Labs' DDSS is designed to be used as a data-sharing platform for application areas including network management and IoT data streaming. This system has the potential to be applied to a campus setting where one or multiple organizations share the system. It is recognized as a contribution to two research projects. First, as a part of the Smart Otaniemi ecosystem, building level intelligence project [62] collects and analyzes energy consumption data of office, university, and residential buildings. After initial data collection, buildings are clustered based on their energy consumption profiles. This clustered data can be used to estimate and support flexible utilization of the energy resources. Nokia Bell Labs' DDSS can be used to manage the collection and initial processing of data for inputs to reporting tools. MegaSense [63, 64] has been another research project which aims to collect and use environmental data for pollution modeling and prediction. It requires data collection from a wide range of environmental sensors including low-cost sensors where machine learning can be used to calibrate data. The DDSS, by Nokia Bell Labs, can be used as a streaming platform for IoT data collection, processing, and communication from sensors to the relevant systems.

In a system such as DDSS, data is not only transmitted in its raw form, but also in several use cases they can be transformed by performing analytics and other operations. In such cases, we consider that there are intermediaries in the data path, which run either in DF or DH depending upon the data processing needs. These intermediaries can write and apply functions to raw data that results in newly processed data consumable by users. Namely, in a building intelligence system, an intermediary user can write a function for alerts in case of unusual events, and another intermediary can write a function for collecting statistics for weekly or monthly reporting. It is also possible that data owners and intermediaries are from two separate organizations.

Data owners should be able to define policies for processing and consuming data to establish complete control. Additionally, to ensure awareness and transparency, any processing performed on raw data by intermediaries should be transparent to data owners. This can certify that if data owners observe any functions applied to their data that are not acceptable, the access policy can be altered, and permissions to process data by intermediaries can be revoked.

Intermediaries who are applying functions on data should also have the right to define the access policies for consumption as well as further processing of the outputs of their functions. Like data owners, they should have the transparency for usage and the right to revoke permissions if needed. Thus, the system should address the control of data resources by both types of users: data owners and intermediaries.

3.3 Implementation design

In this study, we use policy-based reference architecture [53] for access control. The proposed approach builds upon both XACML [57] and NGAC [54] models, enhancing their strengths. By utilizing the XACML reference architecture but employing JSON for policy definition, we achieve greater flexibility, making policy adaptation to specific needs more efficient. This approach focuses on adaptive access control, akin to NGAC, allowing for dynamic policy changes based on evolving user attributes, resource attributes, and environmental factors. Moreover, by combining XACML's one-time decision approach with NGAC's dynamic enforcement principle, particularly for streaming data, we optimize real-time access decisions with dynamic filters, ensuring a more responsive and context-aware access control mechanism. The schematic diagram of the proposed system is shown in Fig. 2.

Identity provider is a microservice that provides a Restful API for authorization and user management. It is designed in a way that members from multiple organizations can share the same data dissemination system. Users of the system can register and authenticate using the identity provider's endpoints. In addition, administrators of organizations can perform user management tasks using the identity provider component. These tasks include the addition or deletion of new users, managing roles in the organization, and assigning or revoking roles to users.

We use JSON Web Tokens (JWT) for user identification. A JWT token has three parts: header, payload, and signature, delimited by a dot: header.payload.signature. The header defines a specific token type and any additional information, such as hashing and encryption algorithms, required to process the JWT token. The payload contains any claims for the entity, usually a user, and any additional information that needs to be passed between the token issuer and the consumer. Lastly, the signature of a JWT token provides integrity protection and is computed using three components: (1) a given secret key, (2) an algorithm for signing, and (3) a base64Url encoded header and payload.

All other parts of the system will be able to accept the tokens from the trusted asserting identity providers. We utilize user information in JWTs to control what data, management, and monitoring capabilities are exposed, and to whom. A JWT token is required to make API requests to any end-points of the system components. Without a JWT token, API end-points are not accesible. So, for end-users to make any request to any of the access control components, they must have a valid JWT token issued by identity provider with their identity and roles. JWT tokens are also needed for system components to communicate with each other. Tokens are issued to users using the end-point from identity provider [GET]/authenticate which has username, password, and organization parameters as input and returns a token with user details if the parameters are valid.

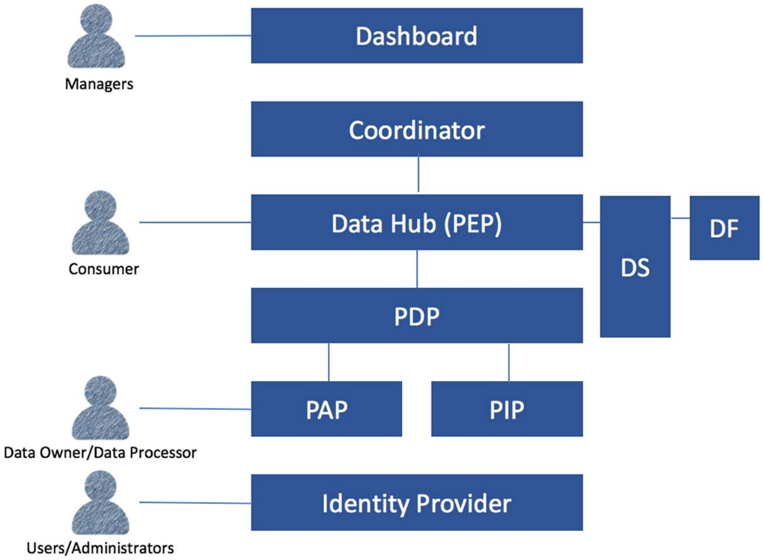

Fig. 2 Access control components added to the Nokia Bell Labs' distributed data streaming system

To save operation time for end-points within the services, we introduce two different kinds of user roles: system roles and organization roles. A system role defines user's relation to the trust zone covering system: its management and use. An organization role defines user's role within an organization under whose mandate he/she uses the system and that has either ownership of some data or has agreement to use data provided by some other organizations. Each service can have its criteria to allow or deny access operation based on the access control mechanism. The first check is performed by each application based on system roles, which change depending on the component the user is interacting with. If a user has the right system role to access the service, they can request to operate using API calls. Figure 2 shows the different system roles for interacting with various components, and we will discuss these roles and interactions in the following paragraphs. After the initial check, requests to API end-points are handled using the checks on organization roles only.

The dashboard offers a comprehensive overview of system activities, accessible to data managers who oversee and ensure the DDSS operates smoothly. Policy Administration Point (PAP) exposes a Restful API for accessing and controlling policies and functions. The two system roles of users governing access to PAP end-points are data owners and data processors. Data owners can define access control policies in JSON format and view how their data is processed and consumed. Data processors with permission to write functions use the PAP endpoint to add metadata of functions and define policies for output resources of those functions. In addition to these two roles, PAP allows requests from other system components defined by system role of system component, for example, to get user policies required to make decisions. Similarly to PAP, PIP also allows requests from other system components to get any additional attributes, such as the type of source, required to make a decision.

DH, which acts as Policy Enforcement Point (PEP) in the system, provides a data catalog that allows data consumers to browse available data sources, submit subscription requests, and monitor their subscribed resources. A user sending request to any of the end-points of DH must have a system role of data consumer. Data catalog is kept open to all users with a system role of data consumer so that data consumers are aware of what sources exist in the system. However, when they request subscription of sources, their organization roles are taken into account when making a decision to accept or deny the subscription request. For example, a user with a data consumer role can make a request to view data catalog from DH but may only subscribe data from Organization A and Organization B based on his organization roles defined in those organizations. If the user tries to subscribe to a source from Organization C, they will be denied and provided with guidance on how to request the required organization roles.

When PEP receives a subscription request, it creates a decision request with all additional attributes for Policy Decision Point (PDP). The PDP component provides a Restful API for access request evaluation. The API end-points of DH are open only for system roles of system components meaning this is only accessible to be used by other system components and not any end-users. The decision request to PDP includes user attributes, requested resources, and desired action. After receiving the decision requests, PDP fetches all required policies from PAP and any attributes required to decide from Policy Information Point (PIP). Currently, PIP supports source attributes like device properties, including its type and make, and can be extended to include other attributes as needed. The PDP component decides for each data resource and sends a response back to the subject, along with the details of which resources can be consumed.

Policy enforcement is divided into two stages when DH receives a response from PDP. This division is important because the streaming data is continuously generated, and it is not possible to make a decision request efficiently for every data point. Instead, a user requests a subscription to a data resource, which is the data stream of a given parameter from a data source. Data owners can define rules on data that can filter data in the permitted streams. We discuss these enforcement and transparency mechanisms in detail as follows.

3.3.1 Static enforcement at subscription creation

The Nokia Bell Labs' distributed data streaming system is designed to be a data-sharing platform with numerous data-producing devices. Hence, it allows consumers to request for multiple parameters from different sources in the same subscription request. The DH sends these requested data resources and user details to PDP and receives a response containing a partial set for which the permission was allowed. The DH will then inform the consumer about this decision and take care of the subscription creation process for those accepted data resources. The response submitted to the consumer may include useful comments about the steps required to make the subscription possible for denied resources. For example, there can be a comment about the required role that consumers can request from administrators in order to subscribe to certain data resources.

3.3.2 Dynamic enforcement at the time of message delivery

When the DH creates a consumer subscription for any accepted data resources, it also creates filters based on the response received from PDP. An example of such filters is to exclude certain records from data streams for consumers with particular attributes. One useful scenario of such filters is when certain records in streams should only be read by consumers within the same organization and should not be visible to other consumers. To implement this, we have used filters that are defined for every user subscription. These filters are checked at run-time for every record received from the data switch. Only the records which pass the filter checks are passed on to the consumers. Figure 3 shows the data flow from DF to consumer.

If a user role or policy is updated in identity provider or PAP respectively, a notification is sent to DH by the system component where update has happened. DH in turn checks for all the existing subscriptions which might be impacted by the change and initiates a decision request for PDP to recheck status of the on-going request. A subscription is ended if user role or policy update denies it in the latest changes. In this way, data owners update policies and system admins can update user roles dynamically and changes are reflected in streams at run time. This follows NGAC's principle of enforcing dynamic user role, organization and policy changes at runtime with a standard set of operations (stream resource) and standard decision function in PDP to evaluate the access control policies.

3.3.3 Transparency and control for data owners

We have also laid the foundation for adding transparency for resource owners as an extension to what XACML and NGAC models offer. When the data owners set up access control policies for their generated data, they are often not acquainted with their later usage and the potential privacy threats. In particular, domains such as healthcare and finance have complex and unpredictable information usage patterns that may not be fully known to the data owner. Enabling transparency becomes a significant enabler in asserting data ownership. It is important to enable transparency as a key component in data access control systems so that the data owners have a clear idea of how their data is processed and used. It should give them a view of how the access policies they have set up are used in the system. By providing a transparent and open view, data owners can understand the policies better and make more rational choices.

Our proposed access control system provides transparency to the subjects who are sharing their data. These subjects include both data owners and data processors. We achieve this by creating a path, along which data is processed, and access is managed from point of origin to point of its usage. This path is created utilizing user-defined policies and functions that are applied to the data. With this transparency mechanism in place, we can ensure that the subjects (data owners) have complete knowledge about their data usage and can see insights about the data: where it is used, how many consumers are

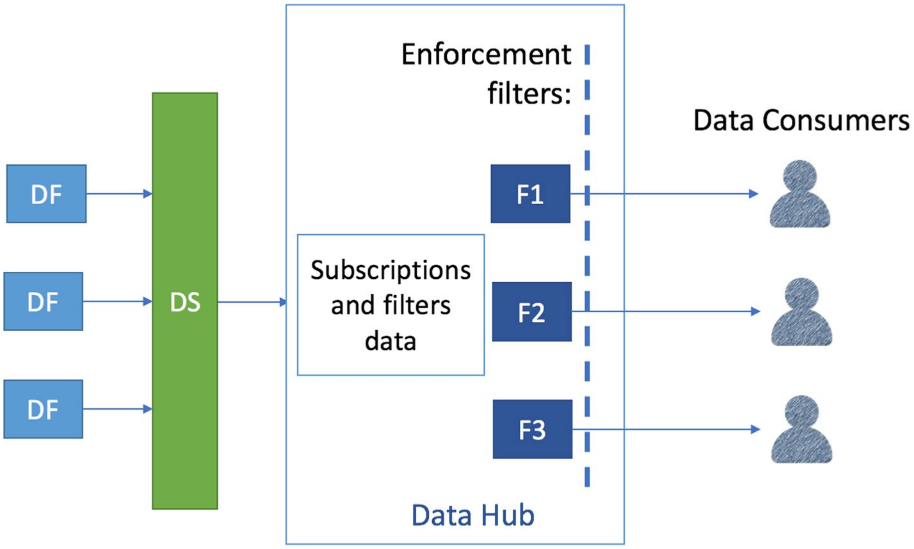

Fig. 3 Dynamic enforcement

consuming it, etc. The service provider on the other hand cannot look into the data content and thus customer (subscriber) privacy is preserved. Apart from that, we assume that the service provider business model respects customer privacy.

As mentioned earlier, every user in the system has a role as each user is configured with a role-based access control model. Furthermore, the users are authenticated and granted JWT token which is checked upon every request for authentication and authorization. For example, every request to the DH is checked for the data consumer system role. The access control policies along with system roles define the limits for the data consumption. The combination of these mentioned mechanisms for access control and data consumption creates a common trust zone in the system. Within that common zone, there are separated zones for data access from different data sources governed by each data owner. Also, the system administration has its own trust zone. These data zones and the administrative zone are separated so that system administrators do not have access to data in data zones without permission from data owner.

To implement the mentioned transparency mechanism, we have added an endpoint in PAP, which provides information about how a data resource is used. The format of this response is a JSON object which shows a hierarchical structure representing stages, in which functions are performed on data resources and their outputs. We can use any graphical tool to create a graph-like structure from this information. On a large scale, system administrators can also use this data to investigate trust structures between users and organizations using networks and graph theory. The analysis can provide useful insights on connections and distributions in data sharing practices, which can be used to design and distribute the systems better. For example, these data-sharing practices can be used to suggest initial policies for new devices based on existing trust networks, or it can be used to analyze and learn the potential privacy risks of policies.

3.4 Experiment

In this section, we present the empirical analysis to evaluate the performance of the proposed access control system. We perform the experiment using four virtual machines (VMs) to simulate the distributed components in our system. VM1 runs Ubuntu 18.04.6 LTS and has an Intel(R) Xeon(R) CPU E5-2680 v3 processor running at 2.50 GHz with 6 core(s) and 16.00 gigabytes of memory. VM2, VM3, and VM4 also run the same OS and processor, but they have different numbers of cores and RAM. Specifically, VM2, VM3, and VM4 have 2 cores and 8 GB of RAM each. The configuration of components is as follows:

- VM1: Hosts Dashboard, Coordinator, and access control components (PAP, PDP, PIP).
- VM2: Hosts DF.
- VM3: Hosts DH.
- VM4: We simulate this as user's machine and send requests to the system using JMeter tool [65].

To assess the proposed access control system, we measure response time(s) of performing different operations with access control (With AC) and without access control (Without AC). The performance impact can determine the overhead of adding access control to the system. The response times are recorded using JMeter tool. For this experiment, we compare the performance of the prototype for three DH's REST API endpoints as shown in Table 1. In all three endpoints as mentioned in Table 1, when access control is used then JWT token is added in the header when calling endpoints. DH endpoint upon receiving the request, first checks for JWT presence and validates it (authentication and authorization checks). If the JWT token is valid, then it returns a response. The endpoint [GET] /sources returns available data source types and their description from DH. The endpoint [POST] /subscription creates a decision request for subscription parameters and sends it to PDP. PDP upon receiving request, fetches all required policies from PAP and other attributes from PIP. Then, based on received policies, attributes, and user role, it decides for each parameter and sends a response back to the DH, along with the details of which parameters can be consumed. DH acts also as PEP in the system and enforces PDP response. Depending on PDP response (allow subscription for all requested parameters, allow subscription for some of requested parameters, or forbid subscription for all requested parameters), subscription is created when permitted. This endpoint returns information about which parameters and sources were permitted, denied, and non-existent. The endpoint [GET] /values checks DH and returns the most recent streamed data for the subscribed parameter.

During this experiment, the success of the desired operation is verified using response codes of access requests. JMeter tests are prepared with user threads ranging from 10 to 100,000 for each operation. The ramp-up period is the time taken for JMeter to start all the threads. For all the tests in this experiment, we have defined a ramp-up period of 1 s. The actual concurrency achieved during the tests is much lower than the defined threads due to the limitations of JMeter, which adds threads at different rates depending on many factors, e.g., system resources, network latency, and connection time. Number of concurrent threads is also lower because

Table 1 DataHub endpoints

Endpoint	Parameters	Description	Response
[GET] /sources		Get information on available source types in the system	Without access control, returns available source types and their description. With access control, returns the sources catalog only if JWT contains valid system role: data consumer
[POST] /subscription	Data type, sources, parameters, time granularity	Create a new subscription for parameters from sources with the given time granularity. A single subscription request can contain many sources and their parameters	Without access control, this endpoint returns a message that the request is received and then creates the subscription. With access control, this endpoint returns a message giving information about which parameters and sources were permitted, denied, and non-existent. Subscription is created for the permitted parameters based on attributes defined in JWT including system and organization roles
[GET] /values	Data type, source, parameter, time granularity	Get the values of the subscribed parameters	Without AC, returns the most recent streamed data for the subscribed parameter. With AC, returns data after checking system and organization roles in JWT

some threads finish before others are started. Figures 4, 5, and 6 show the overhead in average response time for getting information, creating subscription, and monitoring parameter values.

We compiled results for all runs and calculated the average response time for each number of concurrent threads. The concurrency achieved by JMeter varied across the different runs, so the results range from 1 thread to the maximum N concurrent threads recorded in the compiled results. In these experiments, we have categorized the Number of Active Threads (NATs) into disjoint groups and each label in the x axis shows the maximum number of active threads in that group. To illustrate, the distribution of the response extends from the lower to upper quartile values within each set, with a center horizontal line at the median. The lower (upper) whisker extends from the lower (upper) quartile up to 1.5 times the inter-quartile range (IQR = Q3–Q1).The overhead for the implemented prototype has very low impact when few users are making the requests to the endpoints.

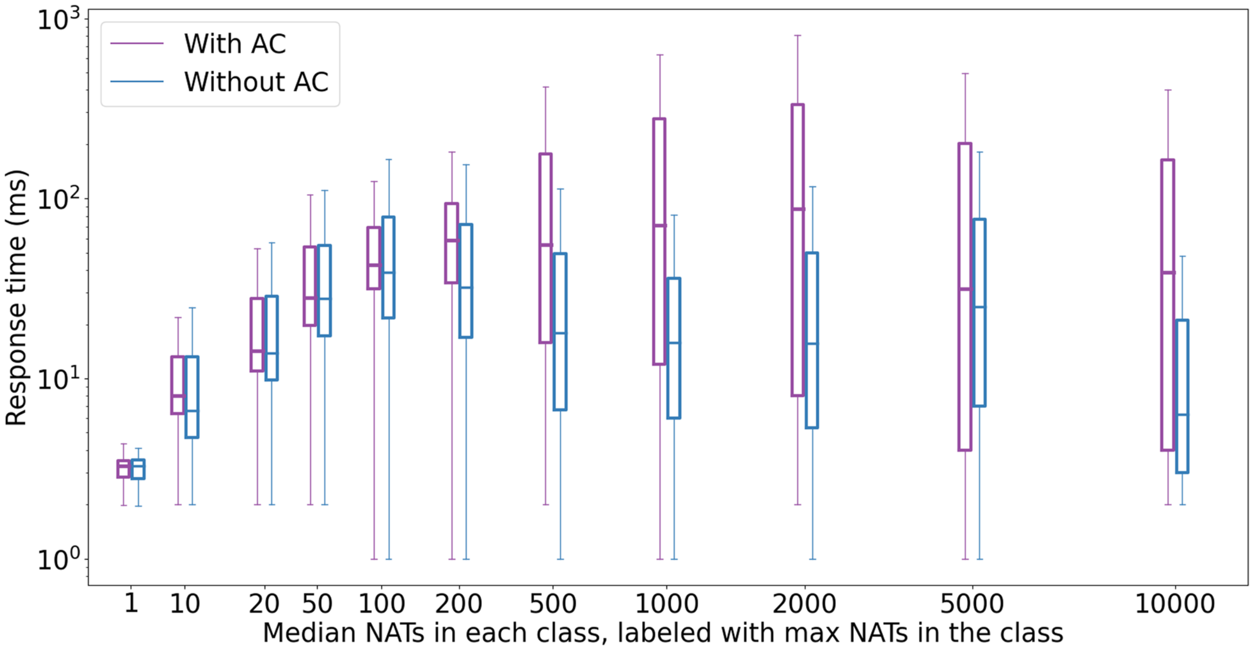

Fig. 4 Response times for getting information

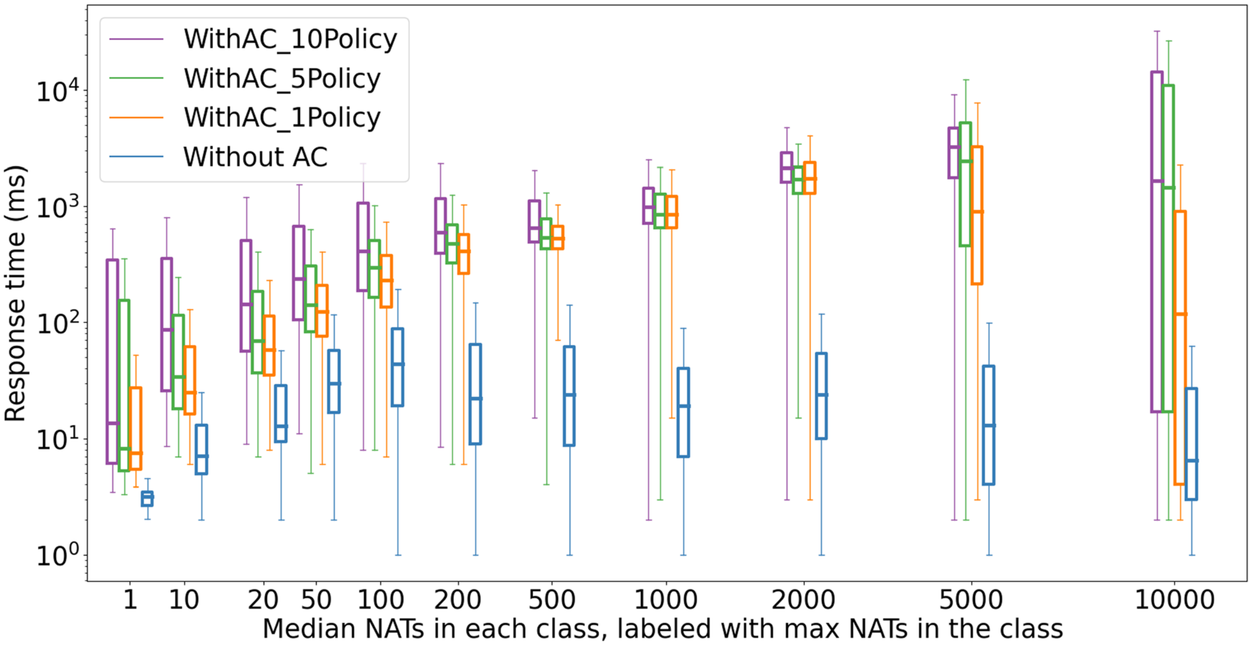

Fig. 5 Response times for creating new subscription

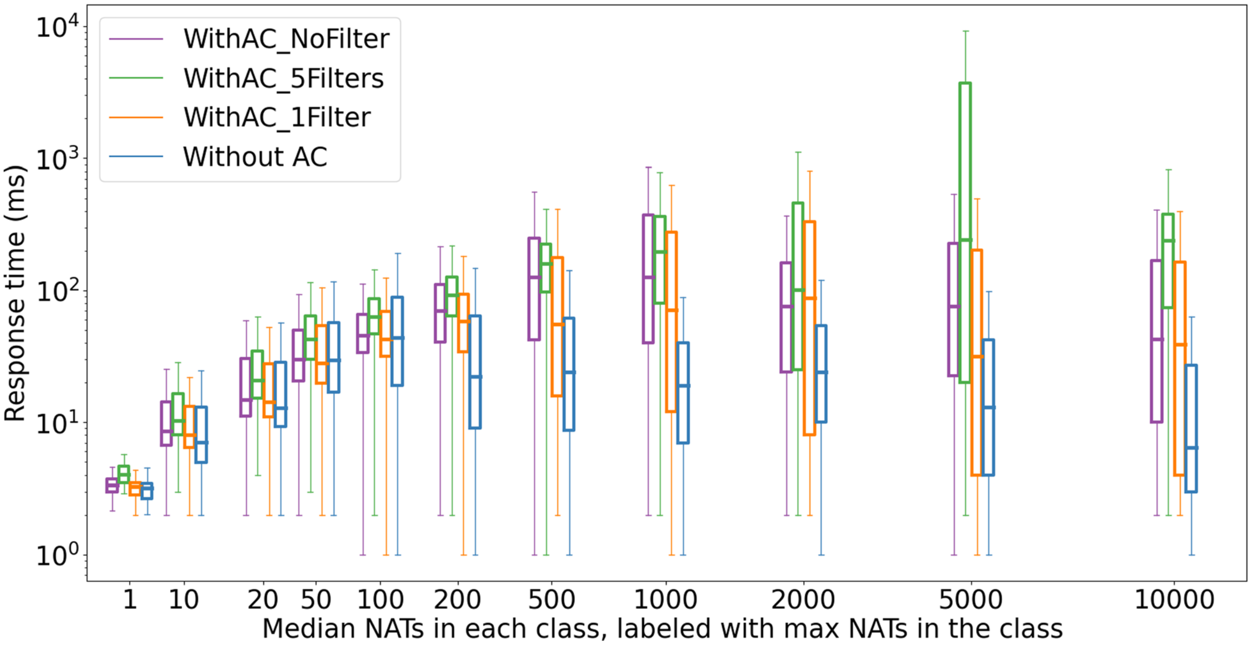

Fig. 6 Response times for monitoring

We observe that with an increasing number of requests, the overhead of adding access control solution increases and this increase is close to linear. The difference is highest for create subscription operation as it requires a request to PDP which in turn requests PAP and PIP for data needed to make a decision. We evaluate the scalability of the prototype by increasing number of policies required to make access decision for a subscription request. We have tested creating subscription using 1, 5, and 10 policies (Fig. 5). We also analyze the difference between average response time on monitoring parameter values if one or more filters are present in a subscription

(Fig. 6). Considering the system performance, we are convinced that the system would meet the performance requirements of upcoming 6G era.

4 Discussion and future work

The paper enhances our knowledge in realizing that big data generated from IoT devices need a well-defined and transparent access control mechanism. Our proposed solution focuses on the requirements and challenges faced in building access control mechanisms in data-sharing systems such as Nokia Bell Labs' DDSS, and insights from this work can provide the foundation for the design and implementation of access control mechanism. We have studied two key research areas. The first area is to design the access control mechanisms for dynamic or streaming data, which is generated from numerous sources in the system. The second key area we considered is to design these systems with a strong focus on how data owners can control and view their data usage. Moreover, data processing, in addition to data consumption, should be managed by access control systems.

By addressing these two areas, we enable Nokia Bell Labs' DDSS [60, 61] to be used as enabling technology for sharing data within organizations or externally with other organizations using the system. Data owners within organizations can define and manage fine-grained access control policies for data consumption. It also provides transparency for data usage and updates data streams dynamically based on updates in access control policies. We have performed experiments to check scalability of the system and our future plans include more detailed study on interoperability and reliability of the system (Section 3.1.4).

We provide experiences and insights as following.

4.1 Data ownership and processing flow

Our proposed access control model ensures that data can be processed in different stages, by different subjects, and each of these stages has different access control rules to govern the further processing or consumption. This allows multiple organizations to cooperate, share, and process data using the same access control mechanism. In future, we can use graph and network analysis tools to analyze data and develop a trust model created based on the history of access control policies and functions. This trust model can be used to generate automated policies when new sources are added to the system.

4.2 Transparency/usability for data processing and consumption

Our proposed system enables data owners to view and control how their data is processed and consumed. It gives them a better picture to make informed decisions for sharing their data. For example, if a data owner views that data from his sensor is used in an aggregation function performed at the building level and wishes not to participate, he can adjust the access control policies to not allow that processing. This meets our requirement of user centric mechanism identified in Section 3.1.2. In future, we can enhance this further by add data lineage information to provide useful insights about data processing and usage.

4.3 Identification of policy components

We have used the standard policy components defined in the policy-based architecture. We have extended the XACML administration component by implementing additional functionality for maintaining intermediaries who perform processing functions on the data. The meta-data of these functions is present in PAP. PDP is designed in the standard way, where it receives a request for a decision and sends back a response with the decision. Our PDP implementation differentiates from standard in the way how it accepts a single request for multiple data resources and provides a response with a sub-set of resources, which are permitted. PAP provides the necessary additional API endpoints for managing the data in different stages by the stakeholders, which are involved in those stages.

Our PEP implementation is done in an already existing DH, where access policies are enforced for data subscription and monitoring requests following NGAC's attribute-based and real-time policy enforcement concepts. The proposed system uses the same architectural components as defined in XACML policy-based access control architecture. However, to avoid additional complexity, the syntax of access policies was kept simple and minimal using JSON as the policy definition format. In future, we can add more extensions based on the XACML specifications. For example, PAP can be extended to use both negative and positive policies. By combining the XACML and NGAC models, we meet the requirement defined in Section 3.1.1.

4.4 Definition of access decisions for data streams

To avoid a decision-making process for every record of streaming data, a decision is made for starting a subscription for a data resource, and filters are added in PEP for that subscription to apply run-time checks. Decision requests can also be made for multiple data resources and responses will contain only the permitted resources, for which subscription can be started. Additionally, decision response provides a useful message to the consumer containing information on why some requested data resources are not permitted and possible steps to obtain access to those.

4.5 Definition of enforcement mechanisms for data streams

When a positive decision for a data resource is received and streaming is started, certain checks are added in PEP for checking individual data points when delivering them to the consumer. These run-time enforcement checks for streams ensure that checks for individual data points do not require connecting to PDP. This approach provides a faster and more efficient way of performing data checks, contributing to the access control requirement defined in Section 3.1.3.

5 Conclusion

In this study, we highlight the crucial role of efficient planning and development in access control mechanisms for data-sharing systems. Transparency, privacy, and data governance emerge as requirements in the context of multi-tenancy and collaboration among organizations. These challenges are expected to be prevalent in the 6G era and demand robust access control solutions. To address these challenges, we propose a novel access control mechanism integrating XACML's architecture and NGAC's attribute-based representation and real-time policy evaluation and enforcement to implement a solution to share data between organizations in a distributed data streaming system based on the publish-subscribe paradigm. Our proposed mechanism enhances fine-grained policies, dynamic enforcement, and transparency, enabling secure and granular access privileges while facilitating efficient data and information sharing. We conducted experiments to evaluate the performance and overhead of the proposed approach. Furthermore, we present a comprehensive review of the existing access control models and their strengths and weaknesses. These insights provide guidance for future designs of data-sharing systems, ensuring the development of trusted and effective solutions in the ever-evolving landscape of data sharing in the 6G era.

Abbreviations

IoT	Internet of Things
BSS	Business Support Systems
SOA	Service-Oriented Architecture
IDSA	International Data Space Association
IDS-RAM	International Data Space Reference Architecture Model
IDS	International Data Space
ODI	Open Data Initiative
MAC	Mandatory Access Control
DAC	Discretionary Access Control
ACL	Access Control List
RBAC	Role-Based Access Control
ABAC	Attribute-Based Access Control
UCON	Usage Control
XACML	eXtensible Access Control Markup Language
NGAC	Next Generation Access Control
DDSS	Distributed Data Streaming System
DF	Data Fetcher
DS	Data Switch
DH	Data Hub
JWT	JSON Web Tokens
PAP	Policy Administration Point
API	Application Programming Interface
PEP	Policy Enforcement Point
PDP	Policy Decision Point
PIP	Policy Information Point
VM	Virtual machine
AC	Access control
NAT	Number of Active Threads
IQR	Inter-quartile range

Authors' contributions

MZ and MS: writing - original draft preparation. MZ: methodology. MZ, KH, VK: System design and architecture. MZ, MS, VK, KH, ST: writing - review and editing. All authors read and approved the final manuscript. The authors declare that they have no competing interests.

Funding

Open Access funding provided by University of Helsinki (including Helsinki University Central Hospital).

Availability of data and materials

The datasets generated and/or analyzed during the current study are not publicly available due to proprietary nature of data.

Data availability

No datasets were generated or analysed during the current study.

Declarations

Ethics approval and consent to participate

Not applicable.

Consent for publication

Not applicable.

Competing interests

The authors declare no competing interests.

Received: 18 February 2024 Accepted: 14 August 2024
Published online: 10 September 2024

References

1. W. He, X. Tian, Y. Chen, D. Chong, Actionable social media competitive analytics for understanding customer experiences. J. Comput. Inf. Syst. **56**(2), 145–155 (2016). https://doi.org/10.1080/08874417.2016.1117377
2. M. Meire, K. Hewett, M. Ballings, V. Kumar, D.V. den Poel, The role of marketer-generated content in customer engagement marketing. J. Mark. **83**(6), 21–42 (2019). https://doi.org/10.1177/0022242919873903
3. R. Carbonneau, K. Laframboise, R.M. Vahidov, Application of machine learning techniques for supply chain demand forecasting. Eur. J. Oper. Res. **184**(3), 1140–1154 (2008). https://doi.org/10.1016/J.EJOR.2006.12.004
4. R. Toorajipour, V. Sohrabpour, A. Nazarpour, P. Oghazi, M. Fischl, Artificial intelligence in supply chain management: a systematic literature review. J. Bus. Res. **122**, 502–517 (2021). https://doi.org/10.1016/j.jbusres.2020.09.009
5. M. Seyedan, F. Mafakheri, Predictive big data analytics for supply chain demand forecasting: methods, applications, and research opportunities. J. Big Data **7**(1), 53 (2020). https://doi.org/10.1186/S40537-020-00329-2
6. First-ever information value index from PwC and Iron Mountain. 2015. https://www.ironmountain.ca/en/resources/infographicsainfogra/i/information-value-index. Accessed 11 Aug 2023

7. C. Perera, C.H. Liu, S. Jayawardena, The emerging internet of things marketplace from an industrial perspective: a survey. IEEE Trans. Emerg. Top. Comput. **3**(4), 585–598 (2015). https://doi.org/10.1109/TETC.2015.2390034
8. International Data Spaces Association. 2024. https://www.internationaldataspaces.org/. Accessed 11 Aug 2023
9. S. Tarkoma, *Publish/Subscribe Systems: Design and Principles* (Wiley, Hoboken, 2012)
10. P.T. Eugster, P. Felber, R. Guerraoui, A. Kermarrec, The many faces of publish/subscribe. ACM Comput. Surv. **35**(2), 114–131 (2003). https://doi.org/10.1145/857076.857078
11. D. Happ, N. Karowski, T. Menzel, V. Handziski, A. Wolisz, Meeting iot platform requirements with open pub/sub solutions. Ann. Télécommun. **72**(1–2), 41–52 (2017). https://doi.org/10.1007/S12243-016-0537-4
12. C.C. Aggarwal, N. Ashish, A. Sheth, The internet of things: a survey from the data-centric perspective. Manag. Min. Sens. Data 383–428 (2013). https://doi.org/10.1007/978-1-4614-6309-2_12
13. M. Richards, *Software architecture patterns*, vol. 4 (O'Reilly Media Inc, Sebastopol, 2015)
14. D. Lu, D. Huang, A. Walenstein, D. Medhi, in *2017 IEEE Symposium on Service-Oriented System Engineering (SOSE)*. A secure microservice framework for IoT (IEEE, 2017), pp. 9–18. https://doi.org/10.1109/sose.2017.27
15. V. Ziegler, H. Viswanathan, H. Flinck, M. Hoffmann, V. Räisänen, K. Hätönen, 6g architecture to connect the worlds. IEEE Access **8**, 173508–173520 (2020)
16. J. Kabbedijk, M. Pors, S. Jansen, S. Brinkkemper, in *Software Architecture*, ed. by P. Avgeriou, U. Zdun. Multi-tenant architecture comparison (Springer International Publishing, Cham, 2014)
17. C. Ngo, Y. Demchenko, C. de Laat, Multi-tenant attribute-based access control for cloud infrastructure services. J. Inf. Secur. Appl. **27–28**, 65–84 (2016). https://doi.org/10.1016/J.JISA.2015.11.005
18. M. Marjani, F. Nasaruddin, A. Gani, A. Karim, I.A.T. Hashem, A. Siddiqa, I. Yaqoob, Big IoT data analytics: architecture, opportunities, and open research challenges. IEEE Access **5**, 5247–5261 (2017). https://doi.org/10.1109/ACCESS.2017.2689040
19. How the open data initiative is fueling the next generation of customer obsession. 2019. https://theblog.adobe.com/open-data-initiativesummit2019/. Accessed 20 Sept 2023
20. DigitalEurope, The voice of digitally transforming industries in Europe. https://www.digitaleurope.org/. Accessed 22 Sept 2023
21. J. Gantz, D. Reinsel, The digital universe in 2020: big data, bigger digital shadows, and biggest growth in the far east. IDC iView IDC Analyze Future **2007**(2012), 1–16 (2012)
22. P. Hacker, B. Petkova, Reining in the big promise of big data: transparency, inequality, and new regulatory frontiers. Nw. J. Tech. Intell. Prop. **15**, 1 (2017)
23. A. Kearney et al., in *World Economic Forum*. Rethinking personal data: a new lens for strengthening trust, vol. 1 (2014)
24. J. Rose, A. Lawrence, E. Baltassis, Bridging the trust gap in personal data. Boston Consulting Group, Boston, MA, USA, Tech. Rep (2018)
25. J. Ukrow, Data protection without frontiers: on the relationship between EU GDPR and amended CoE convention 108. Eur. Data Prot. L. Rev. **4**, 239 (2018)
26. Q.H. Cao, I. Khan, R. Farahbakhsh, G. Madhusudan, G.M. Lee, N. Crespi, in *2016 IEEE International Conference on Communications (ICC)*. A trust model for data sharing in smart cities (IEEE, 2016), pp. 1–7. https://doi.org/10.1109/ICC.2016.7510834
27. D. Roman, G. Stefano, in *2016 2nd International Conference on Open and Big Data (OBD)*. Towards a reference architecture for trusted data marketplaces: the credit scoring perspective (IEEE, 2016), pp. 95–101. https://doi.org/10.1109/OBD.2016.21
28. A.H. Karp, H. Haury, M.H. Davis, From ABAC to ZBAC: the evolution of access control models. J. Inf. Warf. **9**(2), 38–46 (2010)
29. L. Xu, H. Zhang, X. Du, C. Wang, in *2009 International Conference on Networks Security, Wireless Communications and Trusted Computing*. Research on mandatory access control model for application system, vol. 2 (IEEE, 2009), pp. 159–163. https://doi.org/10.1109/nswctc.2009.322
30. R.K. Thomas, R.S. Sandhu et al., in *Proc. 16th National Computer Security Conference*. Discretionary access control in object-oriented databases: issues and research directions (NIST, 1995), pp. 63–74. https://csrc.nist.gov/pubs/conference/1993/09/20/proceedings-16thnational-computer-security-confer/final
31. D. Ferraiolo, J. Cugini, D.R. Kuhn et al., in *Proceedings of 11th Annual Computer Security Application Conference*. Role-based access control (RBAC): features and motivations (NIST, 1995), pp. 241–48. https://csrc.nist.gov/pubs/conference/1995/12/15/rolebased-accesscontrol-rbac-features-and-motivat/final
32. V.C. Hu, D. Ferraiolo, R. Kuhn, A. Schnitzer, K. Sandlin, R. Miller, K. Scarfone et al., Guide to attribute based access control (ABAC) definition and considerations. NIST Spec. Publ. **800**(162), 1–54 (2014)
33. J. Park, R.S. Sandhu, The ucon $_{abc}$ usage control model. ACM Trans. Inf. Syst. Secur. **7**(1), 128–174 (2004). https://doi.org/10.1145/984334.984339
34. Data usage, access and sharing in the digital economy. https://www.ebf.eu/wp-content/uploads/2020/02/Data-economy-EBF-position-paper-Jan-2020.pdf. Accessed 12 Oct 2023
35. V. Estivill-Castro, P. Hough, M.Z. Islam, in *2014 IEEE International Conference on Big Data (Big Data)*. Empowering users of social networks to assess their privacy risks (IEEE, 2014), pp. 644–649. https://doi.org/10.1109/BIGDATA.2014.7004287
36. M. Davari, E. Bertino, in *2019 IEEE International Conference on Big Data (Big Data)*. Access control model extensions to support data privacy protection based on GDPR (IEEE, 2019), pp. 4017–4024. https://doi.org/10.1109/BIGDATA47090.2019.9006455
37. E. Nwafor, A. Campbell, D. Hill, G. Bloom, in *2017 IEEE SmartWorld, Ubiquitous Intelligence & Computing, Advanced & Trusted Computed, Scalable Computing & Communications, Cloud & Big Data Computing, Internet of People and Smart City Innovation (SmartWorld/SCALCOM/UIC/ATC/CBDCom/IOP/SCI)*. Towards a provenance collection framework for internet of things devices (IEEE, 2017), pp. 1–6. https://doi.org/10.1109/UIC-ATC.2017.8397531
38. E. Nwafor, H. Olufowobi, in *2019 IEEE International Conference on Big Data (Big Data)*. Towards an interactive visualization framework for iot device data flow (IEEE, 2019), pp. 4175–4178. https://doi.org/10.1109/BIGDATA47090.2019.9006451
39. B. Carminati, P. Colombo, E. Ferrari, G. Sagirlar, in *2016 IEEE International Conference on Services Computing (SCC)*. Enhancing user control on personal data usage in internet of things ecosystems (IEEE, 2016), pp. 291–298. https://doi.org/10.1109/SCC.2016.45
40. P. Colombo, E. Ferrari, Access control technologies for big data management systems: literature review and future trends. Cybersecur. **2**(1), 3 (2019). https://doi.org/10.1186/S42400-018-0020-9
41. P. Colombo, E. Ferrari, in *Proceedings of the 23nd ACM on Symposium on Access Control Models and Technologies*. Access control enforcement within mqtt-based internet of things ecosystems (2018), pp. 223–234. https://doi.org/10.1145/3205977.3205986
42. B. Carminati, E. Ferrari, J. Cao, K. Tan, A framework to enforce access control over data streams. ACM Trans. Inf. Syst. Secur. **13**(3), 28:1-28:31 (2010). https://doi.org/10.1145/1805974.1805984
43. M. Guerriero, D.A. Tamburri, E. Di Nitto, in *Proceedings of the 13th International Conference on Software Engineering for Adaptive and Self-Managing Systems*. Defining, enforcing and checking privacy policies in data-intensive applications (2018), pp. 172–182. https://doi.org/10.1145/3194133.3194140
44. L. Argento, A. Margheri, F. Paci, V. Sassone, N. Zannone, in *IFIP Annual Conference on Data and Applications Security and Privacy*. Towards adaptive access control (Springer, 2018), pp. 99–109. https://doi.org/10.1007/978-3-319-95729-6_7
45. C.E. da Silva, J.D.S. da Silva, C. Paterson, R. Calinescu, in *2017 IEEE/ACM 12th International Symposium on Software Engineering for Adaptive and Self-Managing Systems (SEAMS)*. Self-adaptive role-based access control for business processes (IEEE, 2017), pp. 193–203. https://doi.org/10.1109/SEAMS.2017.13
46. E.R. Emanuel Onica, H. Mercier, Trust and privacy in development of publish/subscribe systems. (2019). https://aisel.aisnet.org/isd2014/proceedings2019/Society/5. Accessed 11 Sept 2023
47. Gaia-X: A federated and secure data infrastructure. https://gaia-x.eu/what-is-gaia-x/about-gaia-x/.Accessed 15 Aug 2023
48. S. Akther, Gaia-X compatible data flow monitoring in data exchange system. Master's thesis, Tampere University, Finland (2022)
49. P. Colombo, E. Ferrari, Privacy aware access control for big data: a research roadmap. Big Data Res. **2**(4), 145–154 (2015). https://doi.org/10.1016/J.BDR.2015.08.001

50. P. Colombo, E. Ferrari, in *Proceedings of the 23nd ACM on Symposium on Access Control Models and Technologies*. Access control in the era of big data: state of the art and research directions (2018), pp. 185–192. https://doi.org/10.1145/3205977.3205998
51. A. Ouaddah, H. Mousannif, A.A.E. Kalam, A.A. Ouahman, Access control in the internet of things: big challenges and new opportunities. Comput. Netw. **112**, 237–262 (2017). https://doi.org/10.1016/J.COMNET.2016.11.007
52. S. Ravidas, A. Lekidis, F. Paci, N. Zannone, Access control in internet-of-things: a survey. J. Netw. Comput. Appl. **144**, 79–101 (2019). https://doi.org/10.1016/J.JNCA.2019.06.017
53. A. Anderson, B. Parducci, C. Adams, D. Flinn, G. Brose, H. Lockhart, K. Beznosov, M. Kudo, P. Humenn, S. Godik, S. Andersen, S. Crocker, T. Moses, eXtensible Access Control Markup2 Language (XACML) Version 1.0. OASIS (2023). https://www.oasis-open.org/committees/xacml/repository/oasis-xacml-1.0.pdf. Accessed 24 Aug 2023
54. D. Ferraiolo, R. Chandramouli, R. Kuhn, V. Hu, in *Proceedings of the 2016 ACM International Workshop on Attribute Based Access Control*. Extensible access control markup language (XACML) and next generation access control (NGAC) (ACM, 2016), pp. 13–24. https://doi.org/10.1145/2875491.2875496
55. T.H.J. Kim, L. Bauer, J. Newsome, A. Perrig, J. Walker, in *5th USENIX Workshop on Hot Topics in Security (HotSec 10)*. Challenges in access right assignment for secure home networks (2010). https://www.usenix.org/conference/hotsec10/challenges-access-right-assignment-secure-home-networks. Accessed 14 Oct 2023
56. H. Ulusoy, P. Colombo, E. Ferrari, M. Kantarcioglu, E. Pattuk, in *Proceedings of the 10th ACM Symposium on Information, Computer and Communications Security*. GuardMR: fine-grained security policy enforcement for MapReduce systems (2015), pp. 285–296. https://doi.org/10.1145/2714576.2714624
57. R.V. Nehme, H.S. Lim, E. Bertino, in *Proceedings of the Third ACM Conference on Data and Application Security and Privacy*. FENCE: continuous access control enforcement in dynamic data stream environments (2013), pp. 243–254. https://doi.org/10.1145/2435349.2435383
58. OASIS. JSON profile of XACML 3.0 version 1.0. (2020). http://docs.oasis-open.org/xacml/xacml-json-http/v1.0/xacml-json-http-v1.0.html. Accessed 24 Aug 2023
59. P.G. Scaglioso, C. Basile, A. Lioy, Modern standard-based access control in network services: XACML in action. IJCSNS **8**(12), 296 (2008)
60. V. Kojola, S. Kapoor, K. Hätönen, in *Mobile Networks and Management - 8th International Conference, MONAMI 2016, Abu Dhabi, United Arab Emirates, October 23-24, 2016, Revised Selected Papers, Lecture Notes of the Institute for Computer Sciences, Social Informatics and Telecommunications Engineering*, vol. 191, ed. by R. Agüero, Y. Zaki, B. Wenning, A. Förster, A. Timm-Giel, Distributed computing of management data in a telecommunications network (2016), pp. 146–159. https://doi.org/10.1007/978-3-319-52712-3_11
61. N.H. Motlagh, S. Kapoor, R. Alhalaseh, S. Tarkoma, K. Hätönen, Quality of monitoring for cellular networks. IEEE Trans. Netw. Serv. Manag. **19**(1), 381–391 (2022). https://doi.org/10.1109/TNSM.2021.3112467
62. T. Vesanen, K. Piira, R. Lavikka, J. Piippo, H. Biström, L. Kannari, Data Platform for Smart Otaniemi Ecosystem: Requirements, Specifications, and Usage. VTT Technical Research Centre of Finland, VTT Research Report No. VTT-R-00041-21 (2021), https://cris.vtt.fi/files/43219923/VTT_R_00041_21.pdf. Accessed 15 Aug 2023
63. E. Lagerspetz, S. Varjonen, F. Concas, J. Mineraud, S. Tarkoma, in *Proceedings of the 24th Annual International Conference on Mobile Computing and Networking*. MegaSense: megacity-scale accurate air quality sensing with the edge (ACM, 2018), pp. 843–845. https://doi.org/10.1145/3241539.3267724
64. N.H. Motlagh, T. Petäjä, M. Kulmala, S. Tarkoma, E. Lagerspetz, P. Nurmi, X. Li, S. Varjonen, J. Mineraud, M. Siekkinen, A. Rebeiro-Hargrave, T. Hussein, Toward massive scale air quality monitoring. IEEE Commun. Mag. **58**(2), 54–59 (2020). https://doi.org/10.1109/MCOM.001.1900515
65. E.H. Halili, Apache JMeter: A Practical Beginner's Guide to Automated Testing and Performance Measurement for Your Websites (Packt Publishing, 2008). https://books.google.com/books/about/Apache_JMeter.html?id=nX8oKIEvUcYC&redir_esc=y

Publisher's Note