Future Generation Computer Systems 142 (2023) 59–74

Contents lists available at ScienceDirect

Future Generation Computer Systems

journal homepage: www.elsevier.com/locate/fgcs

A secured big-data sharing platform for materials genome engineering: State-of-the-art, challenges and architecture☆

Ran Wang [a,b], Cheng Xu [a,b,*], Runshi Dong [a], Zhenghui Luo [a], Rong Zheng [a], Xiaotong Zhang [a,b,**]

[a] School of Computer and Communication Engineering, University of Science and Technology Beijing, Beijing 100083, China
[b] Shunde Innovation School, University of Science and Technology Beijing, Foshan, Guangdong 528399, China

ARTICLE INFO

Article history:
Received 4 July 2022
Received in revised form 25 October 2022
Accepted 22 December 2022
Available online 29 December 2022

Keywords:
Materials genome engineering
Big data
Data sharing
Blockchain
Merkle Patricia Tree
Secure multi-party computation

ABSTRACT

Materials are the foundation of social development. The vigorous development of big-data technology has brought new opportunities for material research and development, gradually entering the data-driven paradigm. How to safely collect, store and utilize material big-data to realize the design and prediction of advanced materials has essential research significance and value. Many material big-data platforms have been constructed to gather multi-source heterogeneous material data. However, these traditional platforms are hard to realize the safe and efficient circulation and utilization of data. Relying on the national Materials Genome Engineering (MGE) project, we built a secured big-data sharing platform and proposed corresponding data collection, storage, utilization, and security solutions. On the one hand, the blockchain framework working as a 'middleware' provides a standard application program interface for data interaction between participants, and participants do not need to perceive the underlying system framework; on the other hand, it provides a unified management and security mechanism for the platform. In terms of collection, the dynamic container model is used to solve the data normalization problem, thereby improving data quality. In terms of storage, data adaptors store normalized data in different databases for distributed storage and unified scheduling. The platform provides a unified service gateway to schedule all services for data utilization. The secured big-data sharing platform can improve data utilization, promote material data sharing, accelerate material discovery, and serve the data needs of high-throughput computing and the design of new materials.
© 2022 Elsevier B.V. All rights reserved.

1. Introduction

Materials are the foundation of social development and a symbol and milestone of human civilization and progress. The development of materials directly determines the pace of progress in all aspects of society and is crucial to economic development and national security. The early material data infrastructure was an offline database, which provided basic retrieval functions and later evolved into an online database [1]. With the rapid expansion of the big-data industry, the material data infrastructure is also undergoing revolutionary changes using big-data. Massive material data can be analyzed, mined, and utilized, and then the corresponding value can be obtained. Nowadays, data-driven material research and development [2] has been considered the fourth paradigm after empirical science, theoretical science, and computational simulation. Data-driven techniques can significantly shorten the R&D cycle and reduce costs simultaneously. Therefore, more and more countries have started the construction of material data infrastructure [3–5].

The establishment of the Materials Genome Initiative (MGI) [6] has become a turning point in data-driven materials science, and the database has gradually evolved into a data center that provides materials data and fundamental analysis services. Data mining and artificial intelligence have increasingly promoted researchers to use intelligent algorithms in recent years. Therefore, most data centers focus on developing algorithmic workflows that enable researchers to perform data analysis and mining on databases [7,8]. It marks another inflection point in the history of data-driven materials science, with the transformation of infrastructure into a data intelligence platform that facilitates the discovery of new materials. Many material big-data platforms

☆ This work is supported by the National Key Research and Development Program of China (2021YFB3702403), the National Natural Science Foundation of China (62101029), and the Fundamental Research Funds for the Central Universities (06500127).
* Corresponding author at: School of Computer and Communication Engineering, University of Science and Technology Beijing, Beijing 100083, China.
** Corresponding author.
E-mail addresses: xucheng@ustb.edu.cn (C. Xu), zxt@ies.ustb.edu.cn (X. Zhang).

https://doi.org/10.1016/j.future.2022.12.026
0167-739X/© 2022 Elsevier B.V. All rights reserved.

have been established in the worldwide, such as AFLOW [3], Crystallography Open Database (COD) [9], Gangyan·Xincaidao [10], etc., aggregating heterogeneous material data. These platforms mainly adopt a centralized structure, while some acute problems remain challenging to break through, such as safely collecting, storing, and utilizing material big-data to realize the design of new materials.

Data collection has always been a significant problem in the platform construction for multi-source heterogeneous material big-data. Some existing big-data platforms support only one or two types of databases [11,12], whose data set structure is relatively simple and convenient for retrieval and calculation. However, applications could be limited within some specific fields [6]. On the contrary, some platforms could accept various types of heterogeneous data, but they may not guarantee efficient and accurate retrieval and calculation. Therefore, how to standardize multi-source heterogeneous data is a prominent problem in data collection. The Materials Data Curation System [13] uses data and metadata models expressed as Extensible Markup Language (XML) Schema composed by researchers to generate data entry forms dynamically. The Citrination [14] developed a hierarchical data structure called physical information files, which can hold complex material data, ensuring user-searchable and machine-readable. These infrastructures provide a standardized data format to reduce the heterogeneity in the stored data but enable only technical experts to manipulate these formats due to the introduction of complex data structures.

Regarding data storage, how to safely and efficiently manage massive material big data is still a significant problem that is difficult to solve. Some studies build up very general material data repositories that centrally store as much data as possible without imposing strict restrictions on the structure or format, such as [5,15]. However, this faces data security risks of leakage and tampering. At the same time, most material big-data platforms store data of different structures into corresponding databases, such as MySQL, Oracle, DB2, etc. [16,17], which are difficult for data service providers to manage and audit. On the data consumer's side, how to effectively utilize the multi-source heterogeneous material big-data is still an important problem to be solved. Based on the above issues, Muzammal et al. [18], and Yue et al. [19] used blockchain as a data storage scheme for parties who own a data source, which is a potential solution that can facilitate centralized management and audit of the underlying database. But security and efficiency remain vital issues to be solved in present and future research.

The purpose of the material big-data platform construction, on the one hand, is to realize the exchange of data. On the other hand, the main goal is to realize data's efficient circulation and utilization. The Materials Commons [20] provides open access to a broad range of experimental and simulated materials data and allows collaboration through scientific workflows. The Materials Data Facility [16] provides data infrastructure resources and scalable shared data services to facilitate data publication and discovery. However, these platforms share raw data with consumers, which threatens data privacy. At the same time, the centralized structure of most platforms cannot meet the multi-party collaborative computing that is usually required [21]. A common management and service interface is needed to ensure the security and privacy of multi-party data while providing convenient access and computing services.

In this paper, a secured sharing platform for material big-data is described relying on the national Materials Genome Engineering (MGE) project. As the service hinge of MGE's data applications, our secured big-data sharing platform can provide data consumers access to massive material data resources collected from more than thirty research institutions in China. The platform can effectively solve the primary problems faced in collecting, storing, and utilizing multi-source heterogeneous data, improve data utilization, promote service sharing, accelerate material discovery, and serve the requirements for high-throughput calculations and experiments. The main contributions of this paper are summarized as follows:

(1) A dynamic container model-based data collection system (DCS) is constructed to interact with data providers to standardize multi-source heterogeneous data and reduce the cognitive burden and learning cost of the users. Our proposed model has no restriction on the structure of the uploaded data and can standardize the data set into a general schema to improve the system's availability. On this basis, accurate retrieval and efficient computational analysis could be achieved.
(2) A blockchain-based secured management framework is proposed in the platform. The distributed secure storage framework for big-data based on the blockchain can effectively solve management and security problems faced by data storage. On one hand, each participant can flexibly deploy block nodes without changing the underlying database framework. Data providers and consumers can join/exit at any time and realize the unified management of databases with different types. On the other hand, the distributed ledger can ensure the security of data storage and realize data tamper-proof, traceable, auditable, etc.
(3) We analyze and discuss the platform's performance on uploading and retrieval before and after the adoption of the blockchain framework. To the best of our knowledge, there is barely any relevant work on the test of how blockchain impacts platform performance on the existing secured big data sharing platform. Other researchers building similar platforms would benefit significantly from our performance analysis in this paper. Some performance-related data is precious, showing the actual performance benefits of the proposed decisions.

The rest of this paper is organized as follows. Section 2 introduces the state-of-the-arts and challenges of the material big-data sharing platform. The framework of our secured big-data sharing platform (S-BDSP) for material genome engineering are displayed in Section 3. Section 4 introduces the construction details and makes some discussions. Section 5 summarizes the full text and prospects.

2. State-of-the-arts and challenges

With the rapid expansion of the big-data industry, the open sharing, exchange, and circulation of big-data have gradually become a trend, promoting the release of the data value in all walks of life. Finance, electricity, transportation, industrial Internet of Things, and other fields strongly demand building a big-data sharing platform and already have corresponding solutions based on their data characteristics [22–24]. In the field of materials, the development of big-data platforms is not yet mature, and there is a lack of security mechanisms to ensure the safety of material data sharing. Due to the multi-modality, isomerization, and discreteness characteristics of material data [25], some common problems are still challenging to break through. Different types of materials have different compositions and performance concerns. Even the same material often exists in other structural forms in various databases. Severe fragmentation, isomerization, and decentralization of material data make it very difficult to collect, store and utilize. At the same time, the security issue in material big-data sharing is also a vital issue shared by academia and industry. For owners of material data, some sensitive data

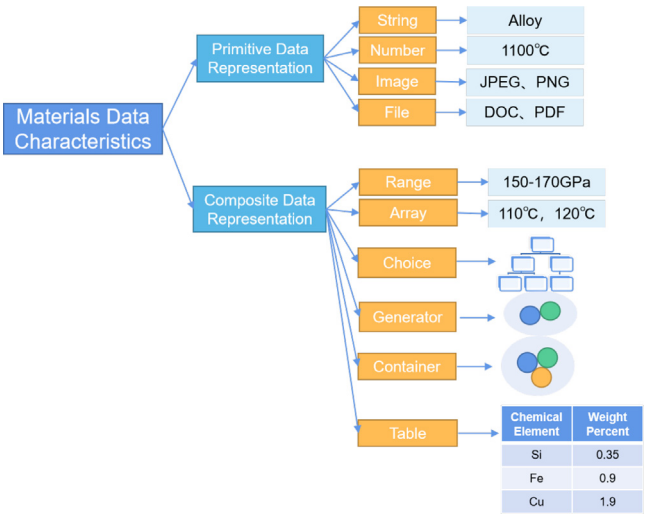


Fig. 1. Classification of materials data characteristics.

represents assets that cannot be easily transferred, resulting in the formation of "data islands". Due to the shortage of high-quality material data, research institutions are not conducive to material science research, which ultimately affects the efficient development of the material industry.

There have been some studies concentrating on the construction of material big-data platforms. Table 1 summarizes the typical schemes of state-of-the-art material big-data platforms from the aspects of data collection, storage, utilization, and security. In the following, this paper will describe the main challenges secured big-data sharing faces from the aspects of the data life cycle and the security mechanisms covering the whole process.

2.1. The characteristics of materials data

The characteristics of materials data are summarized as shown in Fig. 1. Materials data are usually composed of properties with relationships in the abstract. Properties are identified by their names. Values of properties can be described in several different formats called primitive data representations, such as a paragraph of text, a number, a picture, or even files. Composite data representations, such as groups, hierarchies, or tables, describe the relationship between properties. Combining properties defined by different data representations ultimately forms a tree-like data structure. The primitive data representation contains a string, number, image, and file. The composite data representation has range, array, choice, generator, container, and table.

Primitive types are essential components without internal structures. The type String represents a textual description. The type Number represents a numeric value. The type Image and File represent information in image formats and file formats separately. Considering the popularity of pictures in materials data and the requirement for subsequent image processing, we separate Image from File intentionally as an independent data type for high usability. Combinations of built-in types construct composite types. The type Range is composed of a Number and represents an interval value of two numbers; the type Array is composed of an arbitrary built-in type T and indicates that an attribute should take an ordered list of values of T; the type Choice is composed of String and represents the text options that an attribute can take; the type Generator, Container, and Table consist of a collection of fields which are labeled build-in types. They differ in the form of values that a property can take. A property of the Generator can only take one value of some field in the collection. A property of a Container can accept one set of values of all fields in the collection. A property of a Table can take any number of sets of values of all fields in the collection. Table 2 shows some examples of the type declaration form of each build-in type and the corresponding property assignment forms.

Based on the above analysis and summary, severe fragmentation, isomerization, and decentralization of material data make the traditional big-data platforms very difficult to store all types of materials data. When collecting multi-source heterogeneous material data, the conventional material big data platform faces challenges such as insufficient data processing capability, difficulty in unifying data structure, and difficulty in data operation and maintenance, which have created barriers for enterprises to explore the value of data. At the same time, there is no uniform data field definition and database construction standard. Data from different sources will also produce semantic diversity for the description of the same object, which may lead to problems such as table conflicts, value conflicts, and attribute conflicts [25].

2.2. The collection stage

Some material big-data platforms have recognized the importance of *data quality*, such as The Materials Data Facility (MDF) [16] and NOMAD CoE [5]. They use metadata models represented by Extensible Markup Language (XML) schemas to generate data entry forms dynamically. The Citrination platform [14] developed a hierarchical data structure called physical information files, which can hold complex material data, ensuring user-searchable and machine-readable. MARVEL NCCR [4] builds raw input files by converting primary data to AiiDA types through plug-ins. These platforms provide a standardized data format in the collection stage, reducing the stored data's heterogeneity. However, due to the introduction of complex data types and structures, only technical experts can manipulate these formats. This threshold dramatically reduces the operability of the material big-data platforms and goes against the original intention of sharing.

Currently, the industry and academia have little research on the unified data structure and data quality management of the material big-data sharing platform in the collection stage and lack corresponding theoretical research support. For example, most material big-data platforms, such as AFLOW, NOMAD CoE, Materials Project, etc., do not impose restrictions on data uploading and cannot standardize data structure management. Data collection is the first and primary link of a big-data-sharing platform. If the data quality of the sharing platform cannot be guaranteed from the source, unreliable providers may provide biased and inaccurate results to consumers, thereby reducing the availability of shared data.

2.3. The storage stage

Most material big-data platforms adopt a centralized storage structure. For example, a specific mechanism for exchanging and reusing materials data is provided by the Materials Database of the National Institute of Standards and Technology, which accepts data in any format [31]. It is relatively simple and direct, and the "as-is" data submitted by the provider is stored. However, due to its extreme heterogeneity, the data content cannot be directly retrieved or integrated with analysis tools, and only the "as-is" data can be provided directly to the consumer. Material big-data platforms, such as AFLOW, COD, and MARVEL NCCR, centrally store material data in relational databases, file systems, or MySQL. Although they can support simple retrieval and computational

Table 1
A summary of existing typical material big-data platforms.

Name	Description	Collection	Storage	Utilization	Security
AFLOW [3,26,27]	It bases on high flux first principle, and it is one of the largest among many databases. AFLOW has 12 applications including AFLOW π, AFLOW-ML and PAOFLOW, which can screen the structure and properties of materials.	Pauling file, ICSD, Navy crystal lattice database. Collection method is not specified.	Centralized storage in relational database	Retrieval and prediction	–
Crystallography Open Database (COD) [9,12]	COD collects all known "small molecule/small to medium unit cell" crystal structures using an open access distribution model and makes them freely available on the Internet. Cod provides basic upload and search functions.	Check the format of upload files (only supports CIF) using scripts.	Centralized storage in MySQL database.	Retrieving and downloading	Logging, access control, data backup
MARVEL NCCR [4]	Material informatics platform for data-driven high-throughput quantum simulation. Supported by aiida infrastructure.	Convert basic data types to aiida data types through plug-ins to build original input files.	Centralized storage in file system repositories and relational databases.	Retrieval and simulation modeling.	Using a directed acyclic graph to ensure traceability and system robustness
The Materials Data Facility (MDF) [16]	Based on DSpace and Globus systems, MDF operates two cloud hosting services, data publishing, and data discovery. Its function promotes open data sharing, self-service data publishing, and management and encourages data reuse.	A template is provided for collecting data through metadata schema specification.	Distributed storage, stored in databases of different institutions, including local and cloud	Retrieval, data aggregation, and automated analysis.	Identity authentication, access control, disaster recovery backup.
Materials Project [17]	The Materials Project contains a database with a large amount of information (nearly 60000 crystal structures), which can store the results of high-throughput material property calculation. The website also opens a database interface, which allows you to search and filter materials by writing code.	Based on ICSD and other databases.	Distributed storage, stored in different crystal structure databases, including ICSD database, etc.	Data creation, validation, retrieval, download, analysis, and design.	Identity authentication and data integrity verification.
NOMAD CoE [5]	Provide complete input and output file storage of all important computational material science codes, and build multiple big-data services at the top.	Unlimited data uploading, managed by metadata(DOI)	Centralized storage in GPFS file system and MongoDB database.	Retrieve and download(web-based GUI and restful API.	Identity authentication and access control
Open Quantum Materials Database (OQMD) [28]	OQMD is a database based on density functional theory (DFT) to calculate material thermodynamics and structure. It provides an API interface to download the entire database. The research has built a machine learning model calculated by DFT.	According to the given ICSD structure parameters.	Based on ICSD database	Retrieval (web-based GUI and conservative API), data analysis using DFT.	–
Open materials database [29]	COD based calculation database. The open materials database uses a high-throughput toolkit to provide a free open source framework for calculating and analyzing results and storing them in a common and/or specialized database.	Based on COD database.	Based on COD database.	Use high-throughput tools for calculation and analysis.	–
AtSteel [10]	Provide standard data and experimental data of steel, welding materials, and non-ferrous metals, with intelligent matching and material selection algorithms.	Upload data according to the fixed template.	Centralized storage in the public cloud.	Retrieval (providing intelligent data matching service).	Identity Authentication
The Secured Big-Data Sharing Platform for MGE [30]	aims to integrate data resources in the field of materials and establish a material science data system and a material science data sharing service platform that meet different national needs.	Using the dynamic container model, the data sets are automatically merged into containerized data sets.	Adopt "transaction info stored on-chain, and original data stored off-chain."	Retrieval and download, digital identification, secure multi-party computing.	Identity authentication, access control, tamper-proof, security audit, and traceability.

analysis functions, how to centrally and uniformly manage various types of databases and facilitate the upload and retrieval of multi-source heterogeneous data is a complex problem faced by material big-data. At the same time, for the data consumer, the

Table 2
Examples of type declaration forms and property assignment forms.

Classification	Type description	Examples
Primitive type		
String	Any length string.	x = "abc"
Number	Integer or decimal (17 digit precision), with unit.	x = 1
Image	For common picture types, such as JPG, PNG and other formats, you can check the option "allow multiple pictures" to allow uploading multiple pictures.	x = a.png
File	Common file types, such as PDF, word, Excel, json XML and other formats, you can check the option "allow multiple files" to allow multiple files to be uploaded.	x = b.pdf
Composite type		
Range	Numerical range, such as (a, b), or error representation a ±B, a, B are numerical data.	x = (1,2)
Choice	Several string candidates are specified in the template. The options can be grouped. For the time being, only one level is allowed. Click Add option to directly add an item. Clicking Add group is equivalent to adding a level-1 title. You can continue to add several options under this group.	x = "a"
Array	One dimensional array. Select the array type, fill in the field name, and click set to this type. Array name: [1,2,3,4,5,6,7,8]	x = [1,2,3,4]
Generator	The key value pairs corresponding to the name and type are combined. After the combination, the corresponding form is generated by selecting a class.	x = x1 = abc
Container	It can contain all types and can be nested arbitrarily. Drag any type field into the box of the container type to nest successfully.	x = x1 = "abc", x2 = 1,x3 = 2
Table	For tables, six types of data can be added: string type, numeric type, range type, picture type, file type and candidate type. Array type cannot be added. For each column added, select the column type first and click add column to successfully add it.	x = x1 = "a", x2 = 1, x3 = 2, x1 = "b", x2 = 3,x3 = 4, x1 = "c", x2 = 5, x3 = 6

cognitive burden and learning cost is increased, which limits the application field of the material big-data sharing platform.

In response to the above issues of centralized platforms, MDF [16], and Materials Project [17] have adopted a distributed storage solution. However, the data storage structures of various institutions are not uniform, which increases the difficulty for the subsequent data sharing of multi-source heterogeneous materials. In light of this, the blockchain-based storage scheme is a potential solution that can facilitate centralized auditing and management of the underlying database. However, the blockchain represented by Bitcoin was initially designed for digital currency and had extremely high requirements for security. Therefore, complex consensus mechanisms such as proof of work are designed, making the blockchain system slow and challenging to adapt to large-scale data storage. To solve such problems, Muzammal et al. [18] combined the blockchain with the database to build a log-based database application platform, realizing blockchain distribution, decentralization, and audibility. Yue et al. [19] used blockchain as a data storage scheme for parties who own a data source, where data is placed in a specific way to link to blocks. However, due to blockchain's decentralized and open characteristics, storage security cannot be guaranteed. At the same time, storing all the original data on the blockchain is not optimal for the massive amount of material data. The blockchain's consensus mechanism has taken up many resources. If all material data is uploaded to the chain, it will affect the platform's overall performance. Therefore, "on-chain transactions and off-chain storage" will be an alternative to ensure data storage security and performance without compromising platform throughput and computing performance.

2.4. The utilization stage

In using big-data, most material big-data platforms provide data retrieval, download, analysis, and other functions [10]. Usually, users can obtain data by visiting online web pages, which is more convenient and straightforward for the data consumer to use. Furthermore, an application programming interface (API) can be provided for materials informatics applications that require automated access to large amounts of data. For example, the approach adopted by OQMD is to provide an offline version of the database as an interface. Offline access provides the most incredible flexibility and performance. However, it often requires specialized knowledge, such as using Structured Query Language (SQL) or Object Relational Mapping (ORM) to interact with the underlying database. In addition, some material databases provide Digital Object Identifier (DOI) codes to identify data uniquely [32]. However, there is still a gap in the retrieval accuracy and performance of multi-source heterogeneous data to utilize material data efficiently.

One of the significant challenges of material big-data platforms is the computational generation of new components and compounds [33]. Traditional big-data sharing platforms, such as MARVEL NCCR, NOMAD CoE, etc., often aggregate heterogeneous data into different databases. It can be seen that the conventional platform only plays the role of data aggregation and provides point-to-point data transmission services between the platform and the consumer. However, how to use these data to maximize their value after data aggregation and how to develop practical applications based on these sharing platforms are still crucial and challenging problems. Based on the above issues, optimization-based methods [34], such as genetic algorithms and simulated annealing, have been extensively studied, and data-driven methods [35] are beginning to emerge. ALOW [26,27] predicts crystal properties based on machine learning algorithms and provides an open RESTful API to ensure the regular operation of various workflows. However, it is limited to the needs of a single data consumer and cannot satisfy multi-party joint computing.

Various research institutions have gradually increased the demand for joint retrieval and multi-party joint calculation, but existing platforms have not provided relevant functional interfaces. Blockchain-based multi-party computation has been successfully used in other areas. For example, Chen et al. [24] proposed a blockchain-based secure sharing big-data model, combining blockchain and smart contracts with building a reliable data-sharing model that does not require a third party, breaking the current "data islands" and improving data security. Lu et al. [36] proposed a privacy-preserving data sharing scheme based on blockchain and federated learning in the industrial IoT scenario. The blockchain framework can meet the needs of multi-party joint computing and ensure data security.

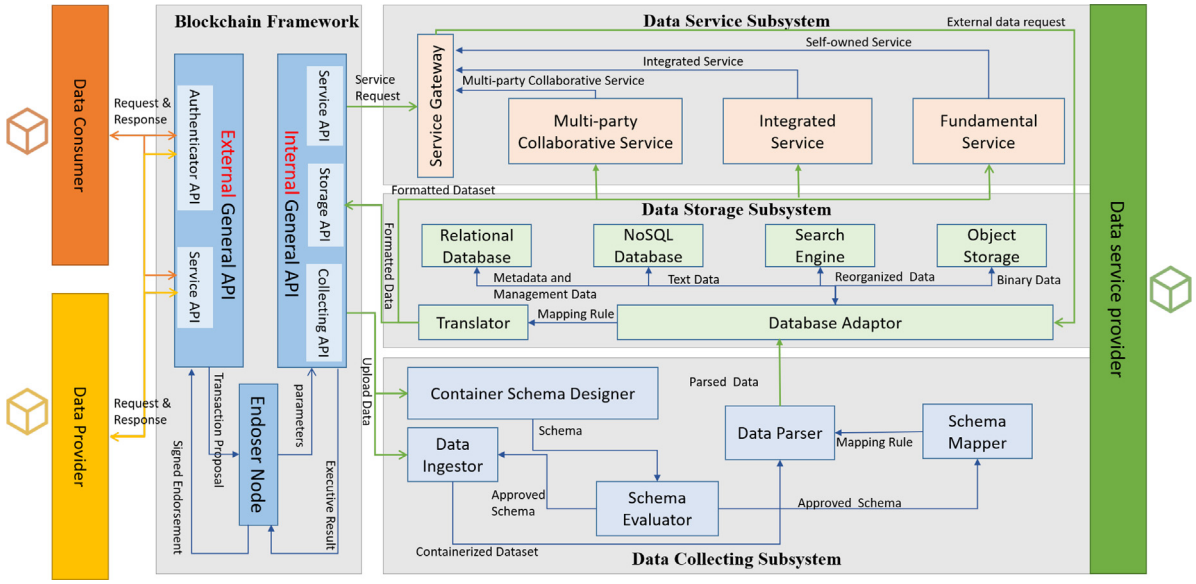


Fig. 2. The framework diagram of our proposed secured material big-data sharing platform.

2.5. Security mechanisms

The material big-data platforms are still relatively weak in terms of security mechanisms. They only provide fundamental security mechanisms, such as authentication and access control. COD [12] provides logging and data backup to ensure auditability of data operations. MARVEL NCCR [4] utilizes a directed acyclic graph to ensure data traceability and system robustness. These mechanisms have good protection capabilities for static data, but they still cannot effectively solve privacy protection problems in data sharing, including data traceability and auditability. During data sharing, the participants may try to infer other's private data from the shared one, leading to the leakage of sensitive data. It is not easy to protect providers' rights and interests.

The emergence of blockchain provides a new direction to solve such problems. The academic community is trying to record the attribution of various data and the access permissions of different users on the trusted blockchain network to ensure that the data is not illegally used [37]. For example, Chen et al. [24] proposed a blockchain-based secured big-data sharing model, where blockchain information is synchronized between various nodes, ensuring the auditability and traceability of data sharing. However, due to the openness and transparency, some studies have introduced additional technologies to the blockchain to ensure data sharing security. Yang et al. [38] proposed a data tamper-proof mechanism based on blockchain. They introduced a cryptographic algorithm to prevent transaction data from being tampered with during user storage, ensuring transaction security and data reliability. Sex. et al. [21] used blockchain-based secure multi-party computation to achieve privacy-protected data sharing. These methods protect the security and privacy of the original data to a certain extent, but they still cannot avoid leakage and tampering in calling results.

Additionally, direct data sharing may result in the data owner losing control of the data, and it is hard to guarantee that there will not be any dishonest participant sharing it with other unauthorized entities. Integrating federated learning into the consensus process of the blockchain realizes the sharing of data models and avoids loss of control over the shared raw data [39]. However, the parameters leakage of the federated learning model leads to the possibility of inferring the original data from the parameters, which remains to be solved urgently. To this end, the academic community has researched the use of secure multi-party computing to enhance the security of federated learning. Wei et al. [40] proposed a new framework based on Differential Privacy (DP), adding artificial noise to the training parameters before the federated model aggregation, thereby protecting the security of the federated learning model parameters. Chai et al. [41] proposed a knowledge-sharing federated learning algorithm based on a hierarchical blockchain. The layered blockchain framework can improve the reliability and security of knowledge sharing. However, it shows weaknesses when dealing with storage problems, whose resource consumption is significant.

3. Secured big-data sharing platform for materials genome engineering

Considering those mentioned above, this paper builds a secured big-data sharing platform for material genome engineering to solve the common and sensitive problems in the existing platforms. It provides solutions to related issues from the collection, storage, utilization of material data, and the security mechanism of the whole process. Our proposed architecture can be capable of data retrieval, multi-party collaborative calculation, and meet the application requirements of material data prediction, modeling, and discovery.

3.1. The overall framework

Based on the underlying data architecture, this paper builds a secured big-data sharing platform for material genome engineering with combining the Hyperledger Fabric consortium chain [42]. It provides an open collaborative environment for researchers to share, retrieve, calculate and analyze data conveniently and securely. The frame diagram of the secured big-data sharing platform is shown in Fig. 2, mainly including data provider, data consumer, data service provider, and the core blockchain framework, providing hub services between the aforementioned three parts.

(1) **Platform Participants:** The secured big-data sharing platform participants for material genome engineering include data providers, data consumers, and the data service provider. The data providers mainly contribute data sources to the platform, and all interactions with the platform are completed through

the blockchain framework. The original data can be uploaded to the uniform storage system of the platform or stored locally at the data provider. The data consumers mainly initiate access requests or service requests to the shared data in the platform, and all interactions with the platform are completed through the blockchain framework. The blockchain framework records all these transactions among the data provider, the data consumer, and the data service provider. The data service provider provides essential services to the data provider and consumer through the internal/external API embedded in the blockchain, so that authorized users can share material data on the platform and work cooperatively to complete the retrieval and analysis of material data.

(2) **Blockchain Framework:** The blockchain plays the role of middleware in the entire platform architecture. As a node on the blockchain, the data provider and consumer send all transaction requests to the blockchain through the external API and then issue transaction tasks to various systems within the platform through the internal API. All users do not need to understand the data service provider's underlying architecture and business processes. The platform's collection, storage, and service systems are transparent to users. The endorsement node of the blockchain executes the smart contract. Then, the internal API transfers the transaction proposal's relevant parameters to the platform's subsystems by calling the smart contract. After that, the endorsement node returns the signature endorsement and proposal execution results to the data provider or consumer. Finally, it generates all transaction results into blocks and synchronizes them to the whole blockchain via the consensus mechanism. The benefit of our proposed blockchain framework is that users do not need to understand the platform's underlying architecture, reducing cognitive load and learning costs. At the same time, the framework provides a more general solution, increases the scalability of the blockchain, and can provide a reference for big data sharing platforms in other industries or fields.

(3) **Data Service Provider:** The data service provider mainly refers to the data collecting, storage, and service subsystems, which provide the data life-cycle services. In the data collecting subsystem, the data ingestor receives the uploaded data and uses the container schema designer to customize the schema to represent the original data set, satisfying the standard data format adopted in the platform. The data storage subsystem stores the original data parsed by the collecting subsystem into different databases and provides the required formatted data to the data consumer and each framework of the data service subsystem. The data service subsystem can provide essential data retrieval, multi-party collaborative computing, third-party integration functions, and other services for data consumers. The service result, that is, the reorganized data set, is stored in the platform data storage system, and the summary information of the service result is stored on the blockchain for subsequent sharing. This bidirectional data flow between the data computation and storage system constitutes a virtuous circle of data sharing and service sharing.

3.2. Data collecting subsystem

Material data providers tend to be diverse, and the raw data are fragmented, heterogeneous, and stored in different formats. If researchers have to manually transfer raw data sets to the infrastructure, the collection process is time-consuming and labor-intensive, reducing their incentive to share data. To this end, our proposed platform developed a data collecting subsystem (DCS) to interact with internal general API of blockchain framework and collect data from data providers. DCS is primarily designed to improve system availability, provide dedicated data collecting tools at the appropriate operational granularity, and automate operations to ease the burden on users. Current implementations of DCS are based on Abstract containers in DCM are designed to have internal structures constructed dynamically from different types of basic layouts. Therefore, DCM provides a way to store, wrap, and exchange data and enables users to customize schemas suitable for the structure of the data. DCM supports customization of attributes and structures. Users can arbitrarily choose attribute names without any restrictions in principle, but practically names in schemas that are publicly available on S-BDSP should follow naming conventions of materials community. The DCM contains the following components: container schema designer, data ingestor, schema evaluator, data parser, and schema mapper. The data ingestor receives the uploaded data and uses the container schema designer to customize the schema to represent the original data set. After the schema is approved, the data provider's raw dataset is normalized and transformed into a containerized dataset, satisfying the standard data format adopted in the platform. Data parser and schema mapper will parse containerized datasets into components such as metadata, textual material data, and binary files, which will be stored in appropriate databases by database adaptor, respectively.

On the basis of material data type, we propose a Dynamic Container Model (DCM) to adapt to the representations of material data. DCM provides a way to store packaging data and allows users to customize schemas suitable for data construction. DCM supports custom properties and structures in principle, and users can choose property names arbitrarily without any restriction. However, these names publicly provided by the platform should follow the material community's naming convention.

DCM consists of two main parts: the container schema, and the container instance. The container schema represents an abstract description of the properties and structure of a material data set. We denote ":" to represent the relationship between the property and the data type. A type declaration expression is denoted as "$x : T$", which means the property x is of type T. A container schema S contains a series of expressions of data-type declaration, denoted as:

$$S = \{x_i : T_i^{i=\{1,\dots,n\}}\} = \{x_1 : T_1, \dots, x_n : T_n\} \tag{1}$$

where x_i indicates the properties and T_i indicates the name of data-types. A container instance represents an abstract description of the data gathered together. It specifies the value of each property and constrained by the data set schema. And the assignment expression "$x = v$" indicates that the value of property x is v at some point. Then, the container instance C could be represented by a series of assignment expressions:

$$C = \{x_i = v_i^{i=\{1,\dots,n\}}\} = \{x_1 = v_1, \dots, x_n = v_n\} \tag{2}$$

Then, a normalized description of a material dataset could be described as a containerized set (S, D), comprising a schema and several instances, where $D = \{C_i^{i=\{1,\dots,n\}}\} = \{C_1, \dots, C_n\}$. It can be seen that DCM uses the template approved by the DCS system to normalize and convert the original data set into a containerized one. The standardized dataset will facilitate subsequent retrieval and computational analysis of material data.

3.3. Data storage subsystem

Material data is an essential resource for developing new materials, so the integrity and availability of material data are critical [25]. To this end, material data collection, storage, and utilization must be transparent, open, and traceable to ensure data quality and achieve reusability. Therefore, to realize the secure management during the overall data life cycle, our proposed S-BDSP adopts the Hyperledger Fabric consortium chain as

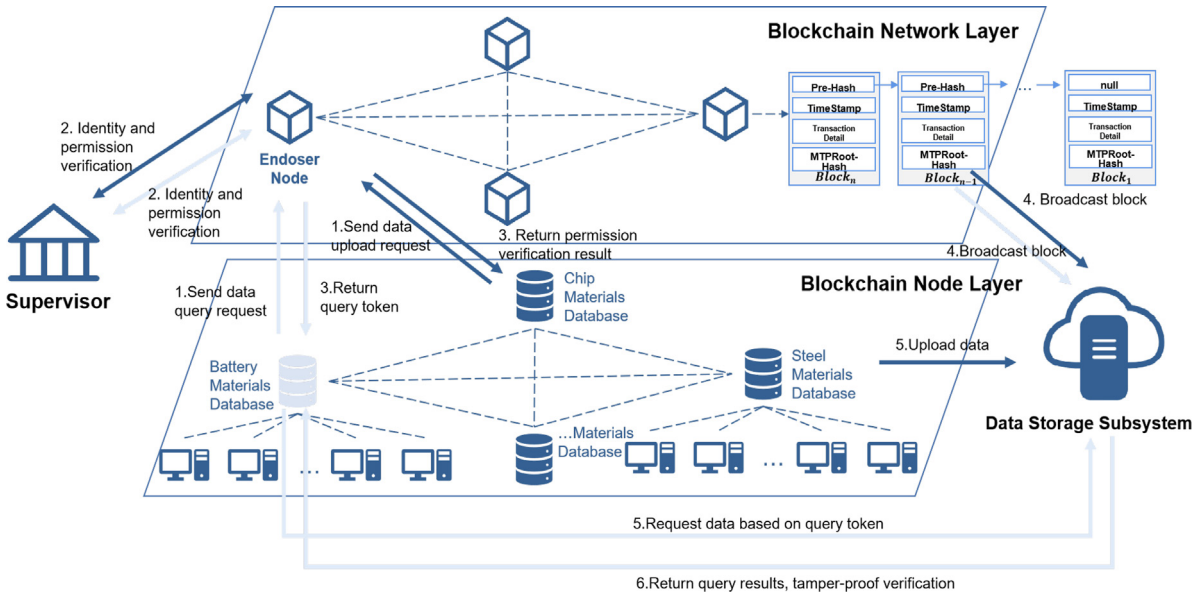


Fig. 3. The secured big-data storage framework based on blockchain.

a middleware of the entire platform architecture to manage the heterogeneous material data, realizing the audit of the whole data processing and ensuring data integrity and availability.

The schematic diagram of the storage architecture for our platform is shown in Fig. 3. Since most material data is sensitive and has a large volume, storing all the data on a blockchain with limited space is resource-wasting and risky. Therefore, we adopt the blockchain to manage and retrieve data due to privacy considerations and storage limitations, using "transaction stored on-chain, original data stored off-chain", which has high security and throughput. The endorsement node in the blockchain (the supervisor is taken as an example of the endorsement node in the figure) is responsible for the user's registration, identity, and authority verification. The upload records of the data provider and the retrieval records of the data consumer will be stored on the blockchain to ensure the transparency and security of the entire process and to achieve data tamper-proof, traceable and auditable. The original data is locally stored or uploaded to the platform by its owner, avoiding the computational pressure on the blockchain. Fig. 3 indicates two main materials data flows: the data uploading flow from data providers and data retrieval flow from data consumers. The two materials data flows form a closed loop. The following takes data upload and retrieval as examples to introduce the overall working mechanism of our blockchain-based secured big-data storage framework.

3.3.1. Data uploading

The blockchain plays the role of middleware in the entire platform architecture. All users do not need to understand the data service provider's underlying architecture and business process. The platform's collection, storage, and service systems are transparent to users. As a node on the blockchain, the data provider sends data uploading request to the endorsement node through the external API of blockchain, including the hash of the uploading data's meta information. The endorsement node checks the data provider's request, determines whether it has permission to upload data, and returns the endorsement result. If the identity verification passes, the endorsement node sends the signature endorsement and proposal execution results to the ordering node, package the request results into a block and broadcast it to each other node on the blockchain. After that, the data provider uploads the data directly to the data service provider. The data collecting subsystem converts the original data into parsed data set, and then uploads it to the data storage subsystem. After the uploading operation is completed, the data service provider return signature to the data provider, informing it that the data uploading is successful. The data provider will initiate another transaction to notify each node that the data has been uploaded successfully. The specific process sequence diagram of data uploading is shown in Fig. 4.

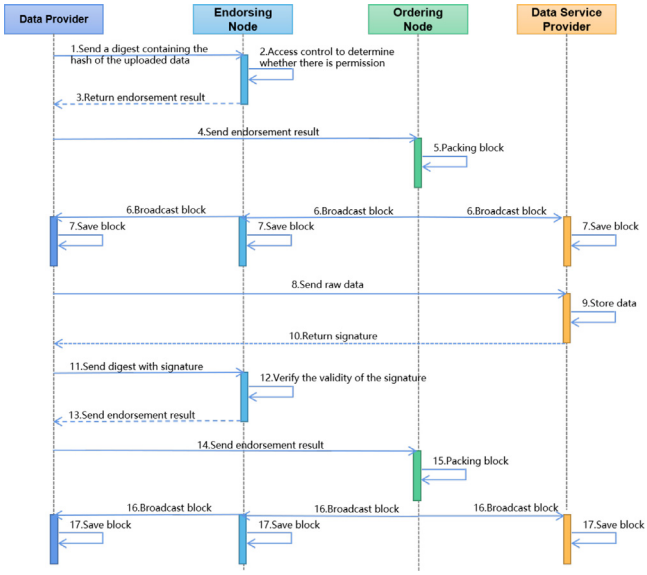


Fig. 4. The workflow diagram of data uploading.

During the whole data uploading process, the data storage subsystem shown in Fig. 2 is responsible for the management of storage technologies which are diverse and optimized for storing different categories of data. The original data set is parsed as components like metadata, textual materials data and binary files by the data collecting subsystem, and these components will be stored separately to appropriate databases by the database adaptor. A relational database is used to store metadata and management data that fit into the relational model. A NoSQL

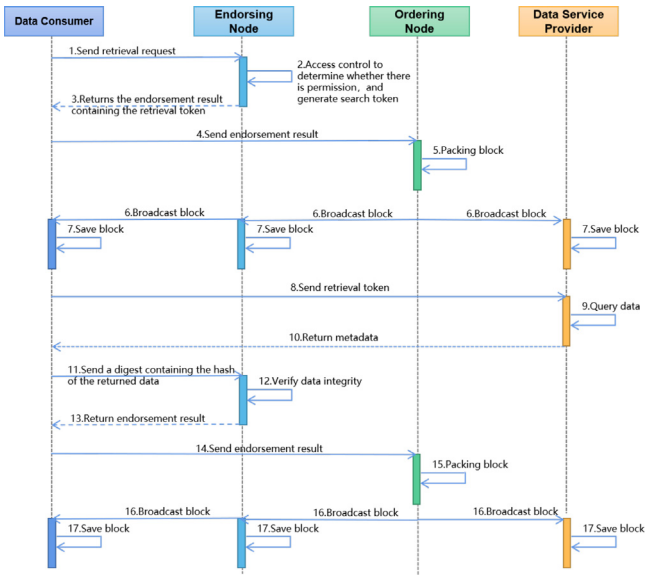


Fig. 5. The workflow diagram of data retrieval.

database is used to store heterogeneous text data that have no fixed schema. All binary data uploaded to the platform are persisted to an object storage. In addition, metadata and text data on the platform are reorganized and indexed in a search engine to enable complex queries. The current implementation of data storage subsystem has adopted the well-known database systems as its backends. Specifically, PostgreSQL, MongoDB, MongoDB̆2019s GridFS, and Elasticsearch are used as the corresponding backends of the relational database, the NoSQL database, the object storage, and the search engine. We are also improving the data storage subsystem to support more database systems.

3.3.2. Data retrieval

The real-time distributed retrieval over the heterogeneous data stored in the platform is the primary requirement of each consumer. In existing scheme that directly interacts with the database, the consumer needs to understand the retrieval methods of various databases for heterogeneous unstructured data. It is not convenient to realize the joint retrieval, and the efficiency is also relatively lower. For our blockchain-based platform, most existing solutions only target on-chain data without considering their correlation with off-chain data. In order not to modify the underlying database and improve the retrieval efficiency, our platform uses the inverted index and Merkle Patricia Tree (MPT) to build the index structure on the consensus chain and forms a mapping relationship between keywords, data block addresses on the chain and database addresses off the chain. In this section, we will introduce the workflow and method of data retrieval.

In terms of *the data retrieval workflow*, the data consumer sends a data retrieval request to the endorsement node by external service API. The endorsement node reviews its identity and determines whether it has retrieval authority. If the identity verification passes, the endorsement node generates a search token and returns it to the consumer. The data consumer sends the endorsement result to the ordering node, packages the data retrieval request into a block, and broadcasts it to each node on the blockchain. After the data service provider saves the block, it sends the search token to the service gateway, which issues a retrieval service task to the fundamental service framework. Then, it finds the summary information of the data set from the blockchain and retrieves the data from different databases

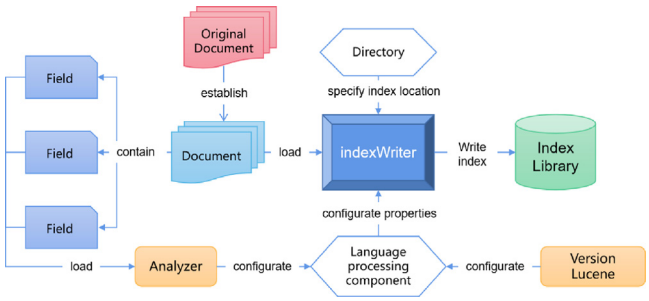


Fig. 6. Create inverted index flowchart.

through the mapping relationship between the blockchain and the database. As shown in Fig. 2, the retrieval result is stored in the data storage subsystem as the reorganized data set. Thus, the retrieval results are translated into formatted data sets through a translator returning to the data consumer. After the data consumer gets the retrieval result, it will initiate another transaction informing each node that the data has been successfully retrieved, and the summary information of the retrieval result is stored on the blockchain for subsequent sharing. The specific process of data retrieval is shown in Fig. 5.

As for *the data retrieval method*, there are mainly three steps: extracting keywords, constructing index structure, and realizing retrieval function. Firstly, we adopt the approach of the inverted index to form a "keyword *v.s.* block address" index structure. For example, as shown in Fig. 6, ElasticSearch (ES) [43] could be used to extract all structured and unstructured data and then create an index. The finally obtained inverted index is shown in Fig. 7, which is used to construct an index structure between keywords and block addresses. Then, another index structure is built through the MPT tree to improve the efficiency of multi-keyword retrieval. The MPT tree could be considered as a fusion of Merkle Tree and Patricia Tree. Based on the anti-tampering hash feature in the Merkle Tree structure, the authenticity and integrity of the retrieval results can be verified. Moreover, Patricia Tree saves data with the same prefix key combination to the same path to free up storage space. Since MPT contains all the index information on the blockchain, its size keeps increasing as the index information accumulates. To reduce the storage cost within the block, MPT introduces a node pointer structure, as shown in Fig. 8. If part of the MPT is not updated in the block, there is no need to keep this part in the newly generated block, but add a node pointer to the last node in the previous block where the MPT has not changed. The MPT in the new block only needs to store updated nodes. Finally, multi-keyword and fuzzy retrieval can be realized through the MPT index. For accurate multi-keyword retrieval, the traversal starts from MPTRoot. When multiple keywords are matched, the value of the path's expansion node or leaf node is returned.

3.4. Data service subsystem

In this section, we outline the implementation architecture of Data Service Subsystem (DSS), which consists of the following components: service gateway, fundamental service, integrated service, and multi-party collaborative service. DSS is transparent to data consumers which just interacts with blockchain through internal general API. The service gateway of DSS provides a unified portal for data consumers to access services. It verifies requests from data consumers and distributes them to corresponding service frameworks.

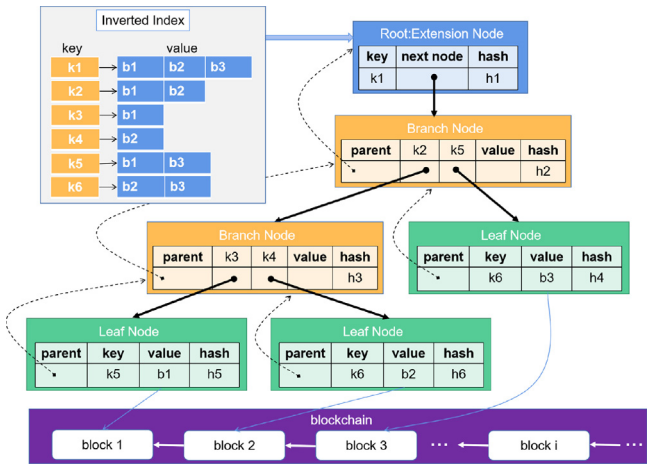


Fig. 7. An MPT diagram of index structure "keyword *v.s.* block address".

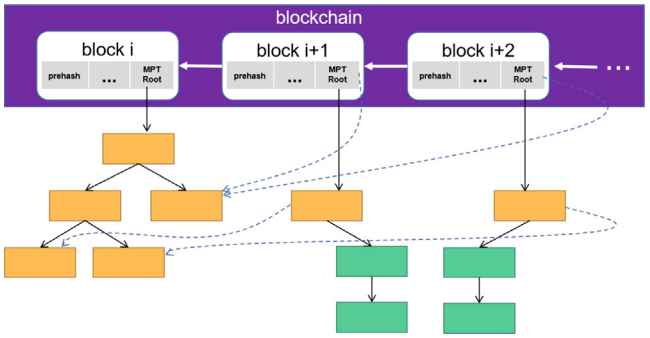


Fig. 8. A diagram indicates the index update process.

3.4.1. Fundamental service

The fundamental service framework provides services that promote data discovery and sharing. It mainly includes search and export services, digital identification services (DIS), and classification and statistics services. The search service provides three kinds of search functions to enable users to make complex queries. The primary search function lets users quickly locate the required datasets through metadata information like data titles, abstracts, owners, and keywords. The advanced search function based on container schemas allows users to impose constraints on data attributes of interest and accurately access the required datasets. The full-text search function has been introduced in *3.3.2*. In terms of DIS, datasets are uniquely identified by the DIS with a DOI that contains information about owners and the location of the underlying dataset. The Association of a digital identifier facilitates the discovery and citation of the dataset. We also provide a classification and statistics service to help users quickly understand the status of materials data on the platform. We divide materials science into different field levels and organize them into a category tree. Statistics information of each field is shown in various visualization methods, including the total amount of data in the platform and a different amount of data in each field with their respective trends in data volume. Other information like the number of visits and downloads of each piece of data, popular fields, and rankings provides users a detailed view to estimate hot data or fields [30].

3.4.2. Integrated service

The integrated service framework is responsible for integrating third-party computing and analysis tools for further research. Third-party online services can directly be integrated into the platform with an access portal in the service gateway and a dedicated API to transfer data. The offline service will provide an introduction portal for users to download and use. At present, the framework under development has integrated several services developed by cooperative teams in our project, such as MatCloud [44] for HTC, OCPMDM [45] for data mining, and the Interatomic Potentials Database [46] for atomistic simulations. There have been some studies using data and services provided by the platform [47,48]. When the framework is fully developed and the integration process standard has been established, the platform will be open to all researchers in the material community and collaborates with them in developing and integrating useful tools that improve data utilization, promoting service sharing and material discovery.

3.4.3. Multi-party collaborative service

Given the complementary relationship between blockchain, federated learning, and secure multi-party computing, this paper adopts a blockchain-based secure computing solution to ensure the security of material data computation among multiple parties. It can ensure that each node has absolute control over its data, and all data calls can be audited in the whole process through the blockchain framework. The multi-party collaborative service framework based on blockchain is shown in Fig. 9. The data consumer initiates a computing request to the platform via the external service API. The request task that carries the model parameters initialized by federated learning is sent to the endorsement node. Calculation scripts for federated learning models are deployed to smart contracts on the blockchain. The data consumer and each participant, namely the consensus node of the blockchain, are responsible for promoting the consensus in the consortium chain. Then, consensus nodes jointly train a global model through federated learning. In the training process, secret sharing is used to exchange model parameters between nodes to prevent the leakage of model parameters and ensure the security of the entire joint training process. Homomorphic encryption technology is used to calculate and update the encrypted model parameters. Once the model is trained, data consumer inputs the parameters to the global model and obtains corresponding results. Finally, the data consumer uploads the calculation results to the consortium chain so that other consumers with the exact computing requirements can obtain the relevant result records quickly, saving the platform's computing cost.

Based on our framework, this section proposes relevant solutions for the collection, storage, and utilization of material data and the security mechanism of the entire process. In terms of collection, problems such as the normalization of the data structure are solved by the dynamic container model. In terms of storage, by building a blockchain architecture based on the underlying databases and adopting the "transaction stored on-chain, original data stored off-chain", centralized management and security audits of different types of databases are realized. At the same time, data leakage prevention, tamper resistance, and traceability can be achieved. In terms of utilization, full-text retrieval of heterogeneous data is realized using inverted index and MPT methods. By using federated learning and secure multi-party computing, collaborative prediction, modeling, and discovery of material properties could be realized.

4. Analysis and discussion

Up to now, more than 13 million pieces of valid material data have been collected through the portal website of the secured big-data sharing platform for material genome engineering (S-BDSP for MGE) [30,49]. The top five areas with the most data

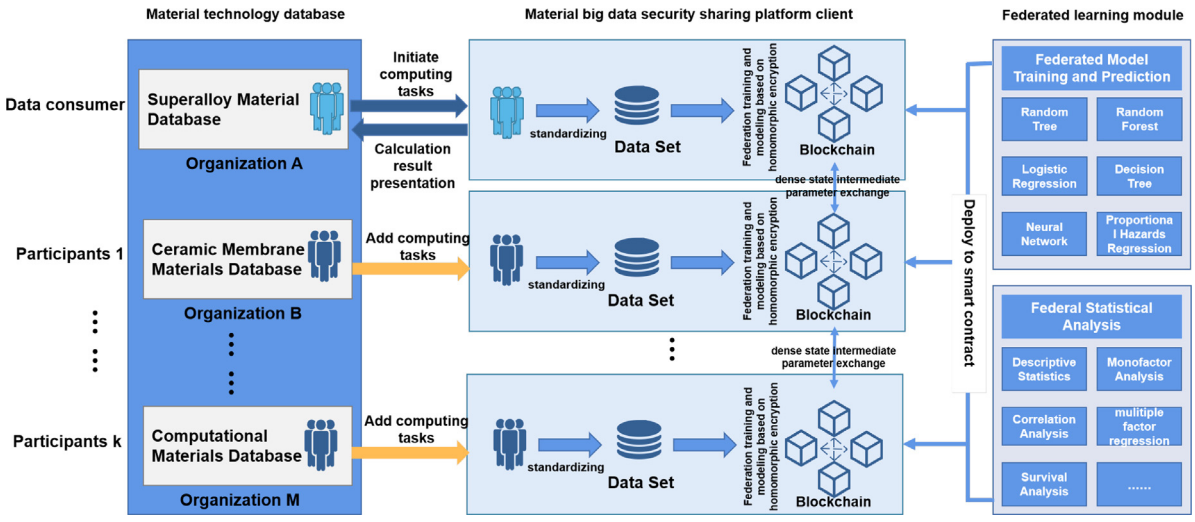


Fig. 9. A framework diagram of secure multi-party computation scheme based on blockchain.

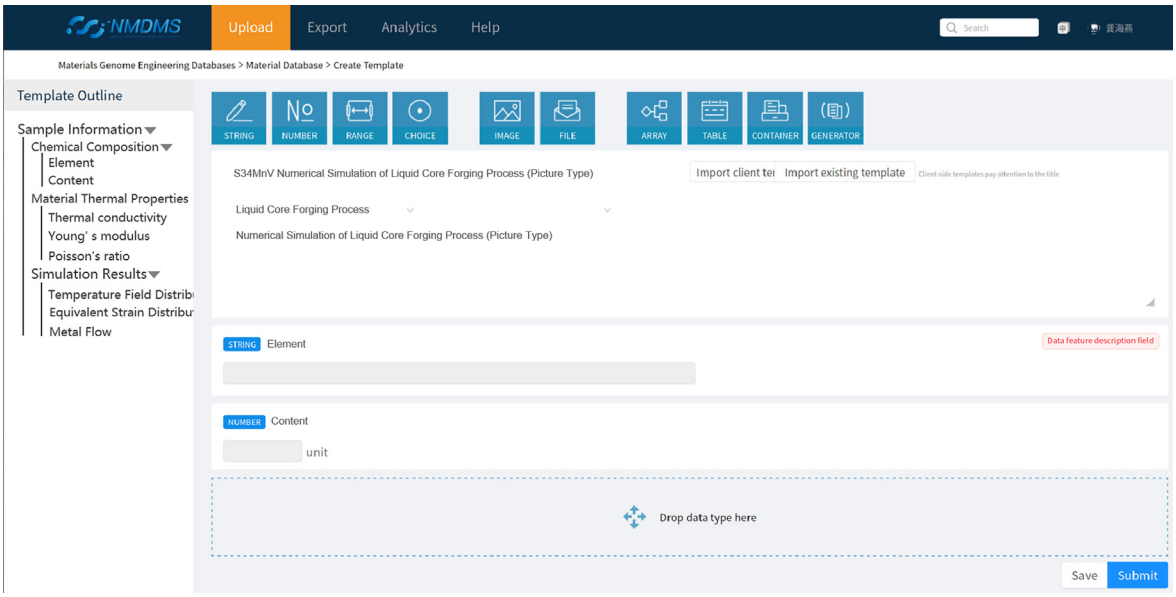


Fig. 10. Graphical user interface of the container schema designer.

are special alloys, materials thermodynamics/kinetics, catalytic materials, first-principles calculations, and biomedical materials. S-BDSP generally provides the solutions for material data collection, storage, utilization, and data-sharing requirements such as data retrieval and calculation among the participants. Data consumers in various fields can also develop their research tools based on the service framework provided by the S-BDSP according to their own sharing needs, jointly predict material properties and develop new materials with other related parties.

4.1. The system function analysis

In terms of data collection modules, since DCM plays a central role in the S-BDSP for MGE, its availability largely determines the performance of the whole platform. To this end, the platform has developed a container schema designer to help users intuitively modify existing schemas or create entirely new ones with built-in types, as shown in Fig. 10. This paper takes the data of shape memory alloy as an example. It shows how to describe the properties and structure of the data through the container schema designer's graphical user interface (GUI), which offers excellent flexibility in creating container schemas. Various schemas can be designed to describe materials in the same field, which will improve the quality of data normalization and make it easier for users to discover and use the data. In addition, DCS provides dedicated data collection tools for each category with appropriate operational granularity, allowing the collection of data sets from providers and automatic normalization into containerized data sets to reduce user workload. Besides, this platform develops a schema evaluator to assess the quality of schemas. With a deep understanding of materials and schemas, assessment experts can correct inappropriate terms and structures in schemas. Approved patterns will be published on the platform.

For data storage, taking data uploading as an example, the S-BDSP can accept data uploaded through web pages or files. After entering the data uploading page, select an existing template in the system, or you can recreate the template and then upload data. The data submission format is either web pages or submitted files, which could be further divided into EXCEL, JSON, or XML files. When submitting via web pages, click the "Submit via the web page", fill in the metadata and related information, and

Fig. 11. The retrieval data representation interface dynamically generated from a container schema. The data representation interface is generated dynamically according to the schema of the example data of the S34MnV liquid core forging process values.

click "Submit". Then, the transaction record of the uploaded data will be stored on the blockchain, including the hash value of the submitted metadata. The original data directly interacts with the underlying database for storage.

The retrieval service provides three modes, i.e., the primary mode, container-based advanced mode, and the full-text mode, enabling users to perform complex queries regarding data retrieval function modules. In the primary retrieval mode, users are allowed to quickly locate the desired dataset through metadata information such as data title, abstract, owner, and keywords. The advanced retrieval mode based on containers will enable users to impose constraints on the data properties of interest and access the required datasets exactly. The full-text retrieval mode allows users to obtain datasets containing multiple keywords in metadata or properties. Each piece of data in the retrieval results will be represented by a visual interface generated by the corresponding schema. As shown in Fig. 11, the details of the S34MnV liquid core forging process values are represented in the generated interface, as the same structure described in the schema. In addition, data sets can be exported to JSON, XML, and excel formats for further study. The result can also be exported with filters, allowing only relevant attributes to be selected. In addition, the platform provides API-based data export for integration services.

In terms of data services, computing and analysis tools can be directly integrated into the service module of S-BDSP through a third-party online service interface and transmitted data with an access portal and a dedicated API in the service gateway. The

Table 3
Experimental equipment configuration.

Configuration	Parameters
vCPU	11th Gen Intel(R) Core(TM) i5-11300H @ 3.10 GHz
Operating System	Ubuntu 20.10
Number of Virtual Machines	5
Memory	16G DDR3 RAM
Language	Python 3.9, Go 1.18
Testing tool	Apache Jmeter and Hyperledger Caliper

framework has integrated several services developed by collaborating teams, such as MatCloud for HTC, OCPMDM for data mining, and an interatomic potential database for atomic simulation. At the same time, based on modules such as federated learning and secure multi-party computing, services, such as multi-party joint prediction of material properties and generation of new materials, are provided. When the framework is fully developed and the integration standard is established, the S-BDSP will open up the related services of multi-party collaborative computing to all researchers in the materials community and cooperate with them to develop and integrate valuable tools to improve data utilization. On the premise of ensuring the security of material data, it promotes the process of material data sharing and material discovery.

4.2. The system performance analysis

In this subsection, our secured big-data sharing platform's performance for materials genome engineering is analyzed based on average latency and throughput. On the one hand, it mainly verifies the platform's performance before and after the adoption of our proposed blockchain framework; on the other hand, we also confirm that our proposed MPT retrieval method can improve efficiency while ensuring security measures. The experimental results prove that applying the blockchain framework makes the platform's performance within an acceptable range, which can fully meet the actual use needs of the system.

4.2.1. Experiment setup

The experimental environment setting needs to be considered from three aspects: blockchain network, equipment configuration, and testing tools. Regarding the blockchain network, each node is launched as a Docker container and then connected to the Fabric network using the Docker Swarm. The blockchain network configuration file was also configured, which defines the network parameters, such as the organizations, peers, nodes, channel name, etc. Specific equipment configuration items and models are shown in Table 3. Regarding testing tools, we used Apache Jmeter to test the platform's performance without blockchain and set up the Hyperledger Caliper-a blockchain benchmark tool to test the platform's performance with blockchain. The configuration file of these tools was configured to vary transaction rates, transaction numbers, and workload containing *uploading* and *retrieval*. Two typical application scenarios as experimental cases are set up to evaluate the critical performance indicators of the proposed system, namely

(1) *Case 1:* The impact of varying transaction rates on the platform's *uploading* performance is evaluated with or without the proposed blockchain framework. Hence, several transaction rates were considered, namely 100, 150, 200, 250, and 300 tps (i.e., transactions per second). The primary purpose of this case is to test the impact of blockchain on system performance containing the average throughput and latency.

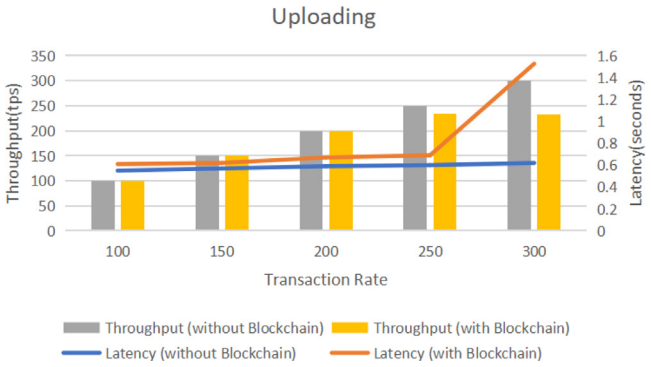


Fig. 12. The impact of transaction rate (tps) on throughput and latency during the *uploading* phase.

(2) *Case 2:* The impact of varying transaction rates on the platform's *retrieval* performance containing the average throughput and latency is evaluated with or without the proposed MPT retrieval method. Hence, several transaction rates were considered, namely 100, 150, 200, 250, and 300 tps (i.e., transactions per second). The case mainly aims at testing whether the MPT retrieval method can improve the platform's performance.

4.2.2. Results and analysis

From the experimental results, it has been found that blockchain impacts the platform's performance to a certain extent but within a reasonable range. Still, according to the practical application of the platform, the transaction rate generally does not exceed 200 tps. When the transaction rate in the experiment is 200 tps, the average throughput during the uploading and retrieval phase has not reached its maximum, and the average latency of uploading and retrieval is about 0.67 s and 0.63 s, respectively. This implies that this specific test environment could handle up to 200 tps without significant network delay. The platform performance is within an acceptable range with introducing the blockchain framework. Compared with the system performance without our proposed framework deployed, the average latency of uploading has risen by only 0.083 s. The increase in millisecond delay is almost invisible to platform users. At the same time, after adding the MTP structure, the average retrieval time drops by more than 50% compared to that of the simple Fabric. Overall, using our proposed blockchain framework, the security performance of the system can be improved without affecting the user experience. Detailed experimental results are discussed below.

As shown in Fig. 12, during the *uploading* phase, compared with the platform without blockchain, the throughput and latency are not apparently influenced when the transaction rate is below 250 tps. The impact of the blockchain framework on the platform performance starts to change significantly when the platform is above 250 tps. That is, the blockchain network could handle up to 250 tps without significant network latency. When the transaction rate goes above 250 tps, the blockchain's throughput decreases as the transaction rate increases and the latency significantly increases. However, in the actual use of the platform, the transaction rate generally does not exceed 200 tps, so after the deployment of the blockchain, the impact of the transaction rate on the platform is acceptable. As the number of concurrent transactions increases, we can easily increase the throughput and latency performance by expanding the hardware resources and network bandwidth without having to adjust the system framework and services.

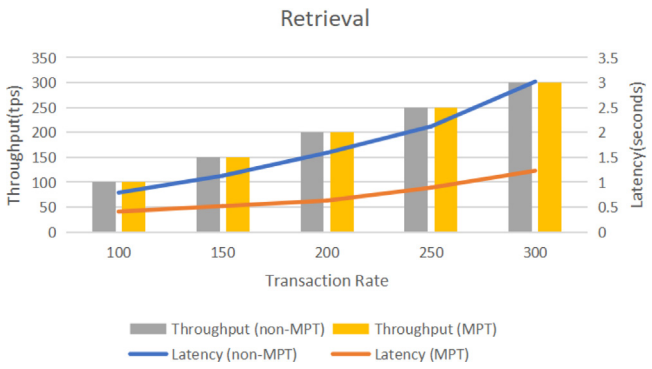


Fig. 13. The impact of MPT retrieval method on throughput and latency during the *retrieval* phase.

During the *retrieval* phase, the blockchain network could handle 300 tps without apparent delay, as shown in Fig. 13. This result illustrates that our actual operating environment did not reach its maximum limit and could support higher transaction rates. At the same time, compared with that retrieval method without MPT (denoted as non-MPT), the average retrieval time of our proposed method drops significantly. From the results, we can verify that the retrieval method based on MPT improves the retrieval performance based on blockchain.

Overall, the following conclusions could be drawn:

(1) During the *uploading* phase, throughput is relatively flat as the transaction rate increases if the platform has already reached at its maximum limit. The higher transaction rate can be supported if the hardware configuration has a higher spec.
(2) During the *uploading* phase, the average latency increases with the increase in transaction rate if the platform has already reached its maximum limit. However, in the actual use of the platform, the transaction rate generally does not exceed 200tps, so after the deployment of the blockchain, the impact of the transaction rate on the platform is acceptable.
(3) During the *retrieval* phase, when the throughput does not reach its maximum limit, compared with simple Fabric, our proposed MPT method's latency growth rate with the increase in transaction rate is not apparent. Although block generation increases the latency, the retrieval method based on MPT improves the retrieval performance. At the same time, after adding the MTP structure, the average retrieval time drops significantly compared to the simple Fabric. From the results, we can verify that the retrieval method based on MPT improves the retrieval performance based on blockchain.

5. Conclusion and prospect

The secured big-data sharing platform for materials genome engineering (MGE) is a national science and technology infrastructure platform. Relying on this, it publishes and provides services through the portal website to support material selection and accelerate material design and optimization, which is significant to national economic construction. In this paper, we analyzed the state-of-the-art and challenges of material big-data platforms and constructed a secured big-data sharing platform framework for materials genome engineering. On the one hand, the blockchain framework working as a 'middleware' provides a standard application program interface for data interaction between participants, and participants do not need to perceive the underlying system framework; on the other hand, it provides unified management and security mechanism for the platform. Its systematic and scientific nature promotes materials genome engineering. Material data plays an increasingly important role in the era of big data, enabling the in-depth development of material science innovation. The cross-integration with information and other fields poses more significant challenges to material data researchers. The analysis and mining-related services of S-BDSP will promote the process of the fourth paradigm, data-driven material research, and development.

The S-BDSP will be further improved and optimized based on the existing framework and related solutions in the follow-up research. For example, the multi-party federal retrieval function would be realized based on heterogeneous data retrieval. At the same time, we will continue to strengthen material data's research and development capabilities in secure sharing to achieve more in-depth material data applications in MGE and other aspects.

CRediT authorship contribution statement

Ran Wang: Conceptualization, Methodology, Writing – original draft. **Cheng Xu:** Conceptualization, Writing – review & editing. **Runshi Dong:** Investigation, Validation. **Zhenghui Luo:** Investigation, Software. **Rong Zheng:** Validation. **Xiaotong Zhang:** Visualization, Supervision.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

Data will be made available on request.

References

[1] Y. Xu, Accomplishment and challenge of materials database toward big data, Chin. Phys. B 27 (11) 118901.
[2] A. Agrawal, A. Choudhary, Perspective: Materials informatics and big data: Realization of the "fourth paradigm" of science in materials science, APL Mater. 4 (5) (2016) 053208, http://dx.doi.org/10.1063/1.4946894, URL http://aip.scitation.org/doi/10.1063/1.4946894.
[3] Aflow - Automatic FLOW for Materials Discovery. URL https://aflowlib.org/.
[4] S.P. Huber, S. Zoupanos, M. Uhrin, L. Talirz, L. Kahle, R. Häuselmann, D. Gresch, T. Müller, A.V. Yakutovich, C.W. Andersen, F.F. Ramirez, C.S. Adorf, F. Gargiulo, S. Kumbhar, E. Passaro, C. Johnston, A. Merkys, A. Cepellotti, N. Mounet, N. Marzari, B. Kozinsky, G. Pizzi, AiiDA 1.0, a scalable computational infrastructure for automated reproducible workflows and data provenance, Sci. Data 7 (1) (2020) 300, http://dx.doi.org/10.1038/s41597-020-00638-4, URL https://www.nature.com/articles/s41597-020-00638-4.
[5] C. Draxl, M. Scheffler, NOMAD: The FAIR concept for big data-driven materials science, MRS Bull. 43 (9) (2018) 676–682, http://dx.doi.org/10.1557/mrs.2018.208, URL http://link.springer.com/10.1557/mrs.2018.208.
[6] J.J. de Pablo, N.E. Jackson, M.A. Webb, L.-Q. Chen, J.E. Moore, D. Morgan, R. Jacobs, T. Pollock, D.G. Schlom, E.S. Toberer, J. Analytis, I. Dabo, D.M. DeLongchamp, G.A. Fiete, G.M. Grason, G. Hautier, Y. Mo, K. Rajan, E.J. Reed, E. Rodriguez, V. Stevanovic, J. Suntivich, K. Thornton, J.-C. Zhao, New frontiers for the materials genome initiative, Npj Comput. Mater. 5 (1) (2019) 41, http://dx.doi.org/10.1038/s41524-019-0173-4, URL http://www.nature.com/articles/s41524-019-0173-4.
[7] S. Caíno-Lores, A. Lapin, J. Carretero, P. Kropf, Applying big data paradigms to a large scale scientific workflow: Lessons learned and future directions, Future Gener. Comput. Syst. 110 (2020) 440–452, http://dx.doi.org/10.1016/j.future.2018.04.014, URL https://linkinghub.elsevier.com/retrieve/pii/S0167739X16308214.

[8] M. Francia, E. Gallinucci, M. Golfarelli, A.G. Leoni, S. Rizzi, N. Santolini, Making data platforms smarter with MOSES, Future Gener. Comput. Syst. 125 (2021) 299–313, http://dx.doi.org/10.1016/j.future.2021.06.031, URL https://www.sciencedirect.com/science/article/pii/S0167739X21002260.
[9] Crystallography Open Database. URL http://www.crystallography.net/cod/.
[10] AtSteel. URL https://www.atsteel.com.cn/.
[11] G. Bergerhoff, R. Hundt, R. Sievers, I.D. Brown, The inorganic crystal structure data base, J. Chem. Inf. Comput. Sci. 23 (2) (1983) 66–69, http://dx.doi.org/10.1021/ci00038a003, URL https://pubs.acs.org/doi/abs/10.1021/ci00038a003.
[12] S. Gražulis, A. Daškevi, A. Merkys, D. Chateigner, L. Lutterotti, M. Quirós, N.R. Serebryanaya, P. Moeck, R.T. Downs, A. Le Bail, Crystallography Open Database (COD): an open-access collection of crystal structures and platform for world-wide collaboration, Nucleic Acids Res. 40 (D1) (2012) D420–D427, http://dx.doi.org/10.1093/nar/gkr900, URL https://academic.oup.com/nar/article-lookup/doi/10.1093/nar/gkr900.
[13] A. Dima, S. Bhaskarla, C. Becker, M. Brady, C. Campbell, P. Dessauw, R. Hanisch, U. Kattner, K. Kroenlein, M. Newrock, et al., Informatics infrastructure for the materials genome initiative, JOM 68 (8) 2053–2064.
[14] J. O'Mara, B. Meredig, K. Michel, Materials data infrastructure: A case study of the citrination platform to examine data import, storage, and access, JOM 68 (8) (2016) 2031–2034, http://dx.doi.org/10.1007/s11837-016-1984-0, URL http://link.springer.com/10.1007/s11837-016-1984-0.
[15] C. Becker, Z. Trautt, L. Hale, NIST Interatomic Potentials Repository, National Institute of Standards and Technology, 2010, http://dx.doi.org/10.18434/M37, URL https://www.ctcms.nist.gov/potentials/. Type: dataset.
[16] B. Blaiszik, K. Chard, J. Pruyne, R. Ananthakrishnan, S. Tuecke, I. Foster, The materials data facility: data services to advance materials science research, JOM 68 (2016) http://dx.doi.org/10.1007/s11837-016-2001-3.
[17] A. Jain, S.P. Ong, G. Hautier, W. Chen, W.D. Richards, S. Dacek, S. Cholia, D. Gunter, D. Skinner, G. Ceder, K.A. Persson, Commentary: The materials project: A materials genome approach to accelerating materials innovation, APL Mater. 1 (1) (2013) 011002, http://dx.doi.org/10.1063/1.4812323, URL http://aip.scitation.org/doi/10.1063/1.4812323.
[18] M. Muzammal, Q. Qu, B. Nasrulin, A. Skovsgaard, A Blockchain Database Application Platform, Tech. Rep., arXiv, 2019, [cs] type: article arXiv:1808.05199. URL http://arxiv.org/abs/1808.05199.
[19] L. Yue, H. Junqin, Q. Shengzhi, W. Ruijin, Big data model of security sharing based on blockchain, in: 2017 3rd International Conference on Big Data Computing and Communications, BIGCOM, IEEE, Chengdu, 2017, pp. 117–121, http://dx.doi.org/10.1109/BIGCOM.2017.31, URL http://ieeexplore.ieee.org/document/8113055/.
[20] B. Puchala, G. Tarcea, E. Marquis, M. Hedstrom, H. Jagadish, J.E. Allison, et al., The materials commons: a collaboration platform and information repository for the global materials community, JOM 68 (8) 2035–2044.
[21] Y. Yang, L. Wei, J. Wu, C. Long, Block-SMPC: A blockchain-based secure multi-party computation for privacy-protected data sharing, in: Proceedings of the 2020 the 2nd International Conference on Blockchain Technology, ACM, Hilo HI USA, 2020, pp. 46–51, http://dx.doi.org/10.1145/3390566.3391664, URL https://dl.acm.org/doi/10.1145/3390566.3391664.
[22] S.R. Pokhrel, J. Choi, Federated learning with blockchain for autonomous vehicles: analysis and design challenges, IEEE Trans. Commun. 68 (8) (2020) 4734–4746, http://dx.doi.org/10.1109/TCOMM.2020.2990686, URL https://ieeexplore.ieee.org/document/9079513/.
[23] C.H. Liu, Q. Lin, S. Wen, Blockchain-enabled data collection and sharing for industrial IoT with deep reinforcement learning, IEEE Trans. Ind. Inform. 15 (6) (2019) 3516–3526, http://dx.doi.org/10.1109/TII.2018.2890203, URL https://ieeexplore.ieee.org/document/8594641/.
[24] Z. Chen, W. Xu, B. Wang, H. Yu, A blockchain-based preserving and sharing system for medical data privacy, Future Gener. Comput. Syst. 124 (2021) 338–350, http://dx.doi.org/10.1016/j.future.2021.05.023, URL https://linkinghub.elsevier.com/retrieve/pii/S0167739X21001734.
[25] L. Himanen, A. Geurts, A.S. Foster, P. Rinke, Data-driven materials science: status, challenges, and perspectives, Adv. Sci. 6 (21) (2019) 1900808, http://dx.doi.org/10.1002/advs.201900808, URL https://onlinelibrary.wiley.com/doi/abs/10.1002/advs.201900808. _eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1002/advs.201900808.
[26] S. Curtarolo, W. Setyawan, G.L.W. Hart, M. Jahnatek, R.V. Chepulskii, R.H. Taylor, S. Wang, J. Xue, K. Yang, O. Levy, M.J. Mehl, H.T. Stokes, D.O. Demchenko, D. Morgan, AFLOW: An automatic framework for high-throughput materials discovery, Comput. Mater. Sci. 58 (2012) 218–226, http://dx.doi.org/10.1016/j.commatsci.2012.02.005, URL http://arxiv.org/abs/1308.5715. arXiv:1308.5715.
[27] S. Curtarolo, W. Setyawan, S. Wang, J. Xue, K. Yang, R. Taylor, L. Nelson, G. Hart, S. Sanvito, M. Buongiorno Nardelli, N. Mingo, O. Levy, AFLOWLIB.ORG: A distributed materials properties repository from high-throughput ab initio calculations, Comput. Mater. Sci. 58 (2012) 227–235, http://dx.doi.org/10.1016/j.commatsci.2012.02.002.
[28] J.E. Saal, S. Kirklin, M. Aykol, B. Meredig, C. Wolverton, Materials design and discovery with high-throughput density functional theory: the open quantum materials database (OQMD), JOM 65 (11) (2013) 1501–1509, http://dx.doi.org/10.1007/s11837-013-0755-4, URL http://link.springer.com/10.1007/s11837-013-0755-4.
[29] Open Materials Database. URL https://openmaterialsdb.se/.
[30] National Meterial Data Management & Service. URL http://mged.nmdms.ustb.edu.cn/analytics/.
[31] Materials Data Repository Home. URL https://materialsdata.nist.gov/.
[32] C. Draxl, M. Scheffler, The NOMAD laboratory: from data sharing to artificial intelligence, J. Phys. Mater. 2 (3) (2019) 036001, http://dx.doi.org/10.1088/2515-7639/ab13bb, URL https://iopscience.iop.org/article/10.1088/2515-7639/ab13bb.
[33] J. Zhou, X. Hong, P. Jin, Information fusion for multi-source material data: progress and challenges, Applied Sciences 9 (17) (2019) 3473.
[34] C. Kim, R. Batra, L. Chen, H. Tran, R. Ramprasad, Polymer design using genetic algorithm and machine learning, Comput. Mater. Sci. 186 (2021) 110067, http://dx.doi.org/10.1016/j.commatsci.2020.110067, URL https://linkinghub.elsevier.com/retrieve/pii/S0927025620305589.
[35] Q. Zhou, S. Lu, Y. Wu, J. Wang, Property-oriented material design based on a data-driven machine learning technique, J. Phys. Chem. Lett. 11 (10) (2020) 3920–3927, http://dx.doi.org/10.1021/acs.jpclett.0c00665, URL https://pubs.acs.org/doi/10.1021/acs.jpclett.0c00665.
[36] Y. Lu, X. Huang, Y. Dai, S. Maharjan, Y. Zhang, Blockchain and federated learning for privacy-preserved data sharing in industrial IoT, IEEE Trans. Ind. Inform. 16 (6) (2020) 4177–4186, http://dx.doi.org/10.1109/TII.2019.2942190, URL https://ieeexplore.ieee.org/document/8843900/.
[37] N. Deepa, Q.-V. Pham, D.C. Nguyen, S. Bhattacharya, B. Prabadevi, T.R. Gadekallu, P.K.R. Maddikunta, F. Fang, P.N. Pathirana, A survey on blockchain for big data: Approaches, opportunities, and future directions, Future Gener. Comput. Syst. 131, 209–226, http://dx.doi.org/10.1016/j.future.2022.01.017.
[38] J. Yang, J. Wen, B. Jiang, H. Wang, Blockchain-based sharing and tamper-proof framework of big data networking, IEEE Network 34 (4) (2020) 62–67, http://dx.doi.org/10.1109/MNET.011.1900374, URL https://ieeexplore.ieee.org/document/9146417/.
[39] D.C. Nguyen, M. Ding, Q.-V. Pham, P.N. Pathirana, L.B. Le, A. Seneviratne, J. Li, D. Niyato, H.V. Poor, Federated learning meets blockchain in edge computing: Opportunities and challenges, IEEE Internet Things J. 8 (16) 12806–12825.
[40] K. Wei, J. Li, M. Ding, C. Ma, H.H. Yang, F. Farokhi, S. Jin, T.Q.S. Quek, H.V. Poor, Federated learning with differential privacy: algorithms and performance analysis, IEEE Trans. Inf. Forensics Secur. 15 (2020) 3454–3469, http://dx.doi.org/10.1109/TIFS.2020.2988575, URL https://ieeexplore.ieee.org/document/9069945/.
[41] H. Chai, S. Leng, Y. Chen, K. Zhang, A hierarchical blockchain-enabled federated learning algorithm for knowledge sharing in internet of vehicles, IEEE Trans. Intell. Transp. Syst. 22 (7) (2021) 3975–3986, http://dx.doi.org/10.1109/TITS.2020.3002712, URL https://ieeexplore.ieee.org/document/9127823/.
[42] E. Androulaki, A. Barger, V. Bortnikov, C. Cachin, K. Christidis, A. De Caro, D. Enyeart, C. Ferris, G. Laventman, Y. Manevich, S. Muralidharan, C. Murthy, B. Nguyen, M. Sethi, G. Singh, K. Smith, A. Sorniotti, C. Stathakopoulou, M. Vukolić, S.W. Cocco, J. Yellick, Hyperledger fabric: a distributed operating system for permissioned blockchains, in: Proceedings of the Thirteenth EuroSys Conference, ACM, Porto Portugal, 2018, pp. 1–15, http://dx.doi.org/10.1145/3190508.3190538, URL https://dl.acm.org/doi/10.1145/3190508.3190538.
[43] A. Yang, S. Zhu, X. Li, J. Yu, M. Wei, C. Li, The research of policy big data retrieval and analysis based on elastic search, in: 2018 International Conference on Artificial Intelligence and Big Data, ICAIBD, IEEE, Chengdu, 2018, pp. 43–46, http://dx.doi.org/10.1109/ICAIBD.2018.8396164, URL https://ieeexplore.ieee.org/document/8396164/.
[44] X. Yang, Z. Wang, X. Zhao, J. Song, M. Zhang, H. Liu, MatCloud: A high-throughput computational infrastructure for integrated management of materials simulation, data and resources, Comput. Mater. Sci. 146 (2018) 319–333, http://dx.doi.org/10.1016/j.commatsci.2018.01.039, URL https://linkinghub.elsevier.com/retrieve/pii/S0927025618300466.

[45] Q. Zhang, D. Chang, X. Zhai, W. Lu, OCPMDM: Online computation platform for materials data mining, Chemometr. Intell. Lab. Syst. 177 (2018) 26–34, http://dx.doi.org/10.1016/j.chemolab.2018.04.004, URL https://linkinghub.elsevier.com/retrieve/pii/S0169743917306949.

[46] L.M. Hale, Z.T. Trautt, C.A. Becker, Evaluating variability with atomistic simulations: the effect of potential and calculation methodology on the modeling of lattice and elastic constants, Modelling Simul. Mater. Sci. Eng. 26 (5) (2018) 055003, http://dx.doi.org/10.1088/1361-651X/aabc05, URL https://iopscience.iop.org/article/10.1088/1361-651X/aabc05.

[47] X.-P. Zhao, H.-Y. Huang, C. Wen, Y.-J. Su, P. Qian, Accelerating the development of multi-component Cu-Al-based shape memory alloys with high elastocaloric property by machine learning, Comput. Mater. Sci. 176, 109521.

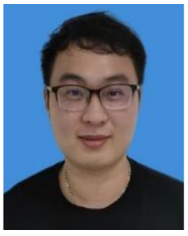
[48] B. Ma, X. Ban, H. Huang, W. Liu, C. Liu, D. Wu, Y. Zhi, A fast algorithm for material image sequential stitching, Comput. Mater. Sci. 158, 1–13.

[49] S. Liu, Y. Su, H. Yin, D. Zhang, J. He, H. Huang, X. Jiang, X. Wang, H. Gong, Z. Li, H. Xiu, J. Wan, X. Zhang, An infrastructure with user-centered presentation data model for integrated management of materials data and services, Npj Comput. Mater. 7 (1) (2021) 88, http://dx.doi.org/10.1038/s41524-021-00557-x, URL http://www.nature.com/articles/s41524-021-00557-x.

Ran Wang received the B.E. degree from the Beijing Information Science and Technology University, China in 2013, and the M.S. degree from the University of Science and Technology Beijing (USTB), China in 2016. She is currently working toward the Doctoral degree at University of Science and Technology Beijing. Her research interests include quantum optimization, distributed security and internet of things.

Cheng Xu received the B.E., M.S. and Ph.D. degree from the University of Science and Technology Beijing (USTB), China in 2012, 2015 and 2019 respectively. He is currently working as an associate professor in the Data and Cyber-Physical System Lab (DCPS) at University of Science and Technology Beijing. He is supported by the Post-doctoral Innovative Talent Support Program from Chinese government in 2019. He is an associate editor of International Journal of Wireless Information Networks. His research interests now include swarm intelligence, multi-robots network, wireless localization and internet of things. He is a member of the IEEE.

Runshi Dong is currently working toward the Master degree at University of Science and Technology Beijing. Her research interests include distributed security and internet of things.



Zhenghui Luo is currently working toward the Master degree at University of Science and Technology Beijing. His research interests include distributed security and internet of things.

Rong Zheng is currently working toward the Ph.D. degree at University of Science and Technology Beijing. Her research interests include distributed security and internet of things.

Xiaotong Zhang received the M.S., and Ph.D. degrees from University of Science and Technology Beijing, in 1997, and 2000, respectively. He was a Professor in the Department of Computer Science and Technology, University of Science and Technology Beijing. His research includes work in quality of wireless channels and networks, wireless sensor networks, networks management, cross-layer design and resource allocation of broadband and wireless network, signal processing of communication and computer architecture.