Potential Identity Resolution Systems for the Industrial Internet of Things: A Survey

Yuzheng Ren, Renchao Xie, *Member, IEEE*, F. Richard Yu, *Fellow, IEEE*, Tao Huang, *Member, IEEE*, and Yunjie Liu

***Abstract*—In recent years, the Industrial Internet of Things (IIoT) came into being. IIoT connects sensors, industrial equipment, products, and staff in the factory, enabling context-awareness and industrial equipment automate control. The identity resolution system is a core infrastructure in IIoT. Similar to the role of Domain Name System (DNS) on the Internet, it is the entrance to the IIoT. The difference between them is their input and output. The input of the identity resolution system in IIoT is an identifier of an object. And the output is the mapping data attached to the identifier, including the product profile, a URL, or the identifier's surrounding environment. However, how to deploy an identity resolution system in IIoT has not yet been conclusive. In this article, we provide a comprehensive survey on the potential identity resolution systems that may be used in IIoT. Firstly, an overview of the identity resolution system is introduced, including a reference framework that can be used to evaluate an identity resolution system. Then we review some influential identity resolution systems based on this reference framework. After that, we make a comparison from the perspective of whether they can meet IIoT requirements and technology selection. Finally, some challenges and broader perspectives are discussed.**

***Index Terms*—Blockstack naming system (BNS), decentralized identifiers (DID), electronic product code (EPC), GNU name system (GNS), handle, industrial Internet of Things (IIoT), identity resolution system, object identifier (OID), ubiquitous ID (UID).**

I. Introduction

With the IoT technology is gradually applied to manufacturing scenarios, the future network is changing from consumption-oriented to production-oriented. According to the 2018 Cisco VNI report, by 2022, the number of machine access will reach 14.6 billion, and the share will reach 51%, which exceeds half of the number of global devices connections [1]. As a result, the concept of smart factories is realized [2], [3], and the Industrial Internet of Things (IIoT) came into being [4]. IIoT connects sensors, industrial equipment, products and staff in the factory, providing context-awareness for intelligent production. So that the industrial processes can be monitored and controlled automatically, optimizing costs, transaction and productivity [5]. Reference [6] has pointed out the scope of IIoT from a functional perspective. They have pointed out that IIoT realize data collection, useful information extraction, information fusion, all information cognition and intelligent decision-making from the bottom-level to the top-level.

IIoT, as a subset of IoT, is proposed for industrial scenarios. IIoT is different from the consumer IoT and has not yet been well resolved. Reference [8] pointed out that what is usually addressed as IoT could be better named as consumer IoT. The main distinction between consumer IoT and IIoT is as follows: 1) The service model is different [5]. Consumer IoT is human-centered, and the purpose of devices interconnection is to improve people's awareness of the environment. In general, Consumer IoT communicates in the form of machine-user and client-server. In contrast, IIoT is machine-to-machine (M2M) communication. 2) Communication requirements are different [3]. IIoT services have higher requirements on scalability, latency, throughput, reliability, robustness, privacy and security. Because the data volume in IIoT is larger, and it is strongly related to enterprise production. 3) The research scope is different. IoT focuses more on designing new communication standards to help devices connect to the Internet ecosystem flexibly and friendly. By contrast, IIoT emphasizes the interconnection between multiple isolated plants and work islands [5].

In order to capture all IIoT connectivity requirements [7], [9], the Industrial Internet Consortium (IIC) provides a new connectivity model, and its stack model is shown in Fig. 1. The lowest layer is the physical layer, where physical signals are exchanged. Frames are exchanged in the link layer using signaling protocols between adjacent participants. The network layer refers to the exchange of packets between non-adjacent participants. Messages are exchanged in the transport layer between participant applications. And in the framework layer, the structured data is exchanged between participant applications with configurable quality-of-service. Above them,

Manuscript received May 1, 2020; revised October 10, 2020 and December 5, 2020; accepted December 12, 2020. Date of publication December 16, 2020; date of current version February 24, 2021. This work was supported in by the National Key Research and Development Program of China under Grant 2019YFB1804403, and in part by the MIIT of China 2019 (Innovative Identification and Resolution System for Industrial Internet of Things). *(Corresponding author: Renchao Xie.)*

Yuzheng Ren is with the State Key Laboratory of Networking and Switching Technology, Beijing University of Posts and Telecommunications, Beijing 100876, China (e-mail: ryz_95@163.com).

Renchao Xie, Tao Huang, and Yunjie Liu are with the State Key Laboratory of Networking and Switching Technology, Beijing University of Posts and Telecommunications, Beijing 100876, China, and also with the Department of Future Network, Purple Mountain Laboratories, Nanjing 211111, China (e-mail: renchao_xie@bupt.edu.cn; htao@bupt.edu.cn; liuyj@chinaunicom.cn).

F. Richard Yu is with the Department of Systems and Computer Engineering, Carleton University, Ottawa, ON K1S 5B6, Canada (e-mail: richard.yu@carleton.ca).

Digital Object Identifier 10.1109/COMST.2020.3045136

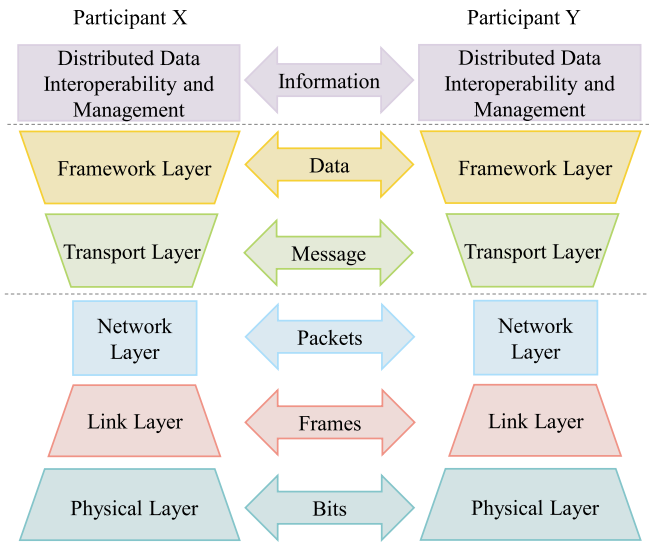


Fig. 1. The connectivity stack model proposed by IIC [7].

outside the scope of connectivity, the distributed data can be interoperated and managed.

The identity resolution system is a core infrastructure in IIoT and works in the framework layer. Similar to the role of Domain Name System (DNS) on the Internet, it is the entrance to IIoT. The difference between them is their input and output. The input of the DNS is a domain name and the output is its IP address. While the input of the identity resolution system is an identifier, and the output is the mapping data attached to the identifier. For example, the output may be the product profile, the URL of the server storing the product information or method, or the identifier's surrounding environment.

The identity resolution system is important for IIoT. It facilitates the intercommunication between multiple isolated work islands by registering, managing and resolving identifiers of objects. A manufacturing process can be described via object and events identifying, which is part of the realization of digital twins [10]. Also, it supports many upper-level applications for effective business processes. The typical applications include product life-cycle management [11], [12], supply chain management and traceability [13]–[17], smart logistics [18]–[20], etc.

However, to the best of our knowledge, there is no identity resolution system specifically designed for IIoT. One available solution is to transplant the identity resolution systems in other scenarios to IIoT. Currently, there are already several published surveys cover different aspects related to identity resolution systems. Reference [21] survey and classify the various representations of digital identities. It mainly focuses on different identifier format schemes. Reference [22] mainly discusses the components of identity modeling, management and requirement matrix from a methodological perspective. Paper [23] discusses a variety of identity management solutions from the perspective of authentication to overcome authentication challenges in resource-limited IoT. References [24], [25] summarize several identity resolution systems from the perspective of supporting IoT. However, these studies are relatively simple in this field and only introduce these systems as a whole, without comparing their pros and cons. And of course, they have no responsibility to discuss whether these systems are suitable for IIoT. Reference [26] surveys blockchain-based identity management systems. This research mainly focuses on discussing the impact of emerging blockchain and the rise of blockchain-based identity solutions.

As far as we know, there is no existing thorough survey of potential identity resolution systems in IIoT. To fill this gap, in this article, we discuss the requirements that the identity resolution systems need to meet in IIoT. And we investigate the state of the art standards and papers of existing important identity resolution systems and fetch common parts from them to build a general reference framework for describing an identity resolution system. Then, we use this framework to survey several important systems. And further, we discuss the fitness of these systems in IIoT and their technology selection. The specific contributions throughout this article are as follows.

- The importance of the identity resolution systems in IIoT is presented. And we discuss the design principles of identity resolution systems for the IIoT. Then a general framework based on key functions for evaluate identity resolution systems is given.
- Some core identity resolution systems are surveyed based on the proposed function-based framework, including the overview, identification, resolution, security and compatibility.
- The comparisons of the proposed systems are given. The systems are first examed based on the principles, and whether they are suitable in IIoT is discussed. Further, each system is compared from the perspective of technology selection.
- The challenges faced in this field and some broad perspectives are discussed.

The rest of this article is organized as follows, as shown in Fig. 2. The overview and background are provided in Section II. In this part, we discuss the importance of identity resolution systems in IIoT and the design principles in sequence. Then we provide a reference framework that can be used to describe identity resolution systems, followed by a rough classification of existing systems. Next, a comprehensive survey of systems based on the provided framework is proposed in Section III. In Section IV, a principle-based comparison is given to discuss whether each system is suitable for IIoT. Since the suitability of the system for IIoT is determined by their supporting technology, a comparison based on technology selection is given later. Combining the gap between existing systems and IIoT, the challenges and future research directions in this field are discussed in Section V. In Section VI, some broader perspectives are presented. And a brief conclusion is given in Section VII.

II. Overview of the Identity Resolution System in the Industrial Internet of Things

Before introducing the specific system, in this section, we discuss the overview of this field. First, we present why

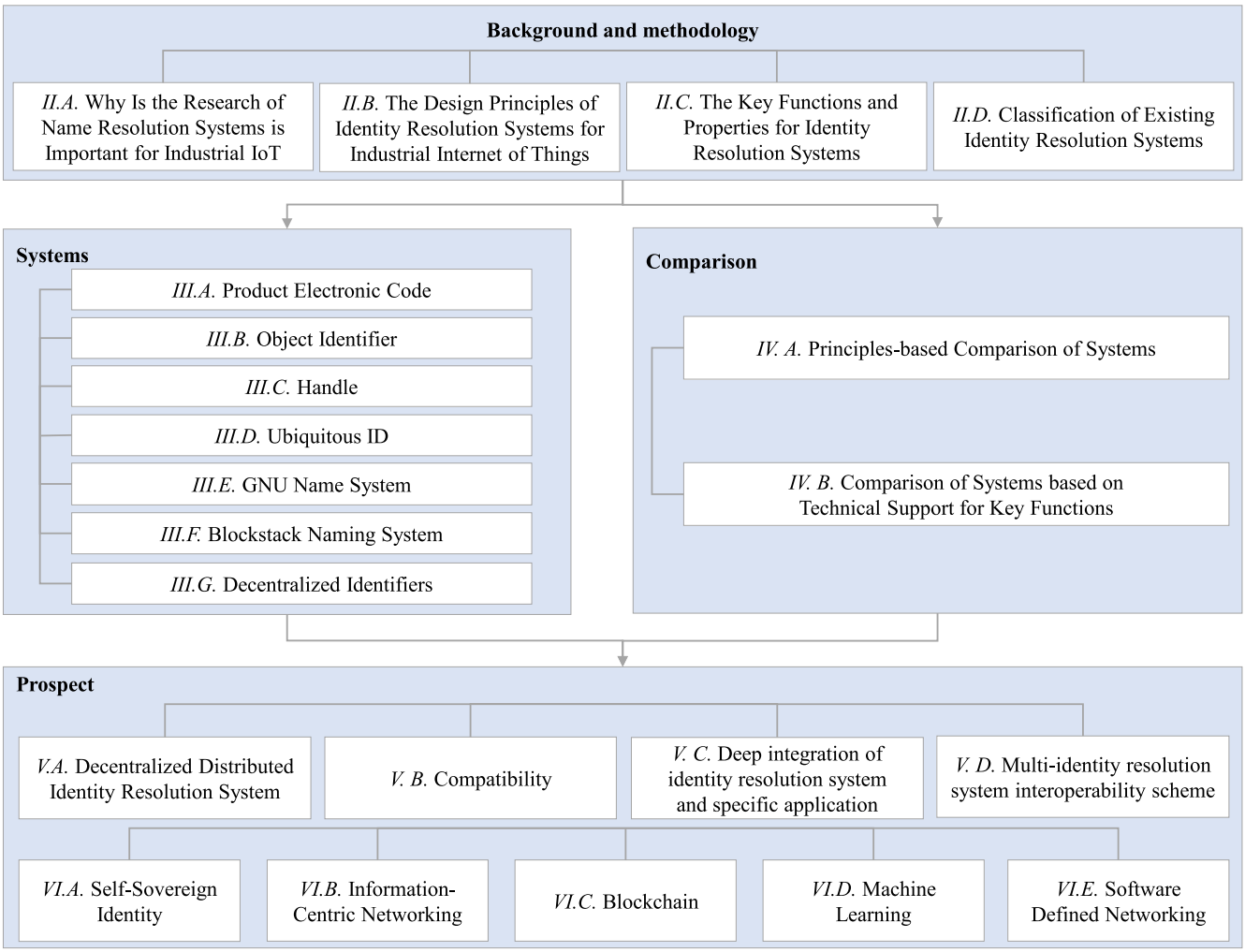


Fig. 2. Road map of this article.

identity resolution systems are so important for IIoT and their role in the IIoT. Then we discuss the design principles, which can be used to exam whether the mentioned systems are suitable for IIoT. Next, we summarize the key functions and properties for identity resolution systems, which constitutes a general framework for describing an identity resolution system. And this framework is also used to compare the surveyed system from the perspective of technology selection. At the end of this section, the coarse classification of existing systems is provided to help readers have a rough understanding of this field.

A. Why Identity Resolution Systems Is Important for the Industrial Internet of Things

The identity resolution system is the entrance to IIoT and its service model is similar to DNS. The similarities and differences between them are shown in Table I. DNS is a core service of the Internet. As a distributed database that maps domain names and IP addresses to each other, DNS can make it easier for people to access the Internet without having to remember complicated IP strings. Similarly, the identity resolution system is a framework layer service in IIoT, providing a mapping relationship between identifiers and corresponding information.

TABLE I
THE SIMILARITIES AND DIFFERENCES BETWEEN DNS IN THE INTERNET AND IDENTITY RESOLUTION SYSTEMS IN THE IIOT

	DNS in the Internet	**Identity resolution systems in the IIoT**
Input	domain names	Identifiers
Output	IP addresses	Information corresponding to the identifier (IP addresses, URLs, product profiles)
Data volume	High	Very high
Criticality	Not stringent	Mission critical (delay, security, privacy)
Intelligence	Medium	High (need to adapt to different performance levels and different types of resolving results required by different applications)

The construction of an identity resolution system can bring many benefits to IIoT. First of all, it is part of the realization of digital twins [10]. Digital twin maps physical space objects

to digital space [27] to realize concise production evaluation. The most relevant topics of the creation of the digital twin can be summarized as identification, data management, digital twin models, digital twin information, human-computer interface and communication [28]. In order to realize the digital twin, some related standards have been proposed. Including philosophy for describing equipment and procedures [29], a standard describing structures [30], XML-based modeling of production environments [31], [32]. The implementation of the identity resolution system provides a unique identifier for each physical object throughout its life cycle. In some digital twin reference models, the identity resolution system appears as a supporting part [11], [33]. It helps build spatiotemporal relationships between multiple digital twins, which is more valuable than a single digital twin. Because this relationship can provide object optimization among multiple systems [34].

Moreover, identity resolution systems can facilitate the intercommunication between multiple isolated plants. Nowadays, division and cooperation have become increasingly detailed among manufacturing enterprises, resulting in relative independence among different departments and companies [35]. As stated in [5], [8], one of the goals of IIoT is enabling communication and understanding between multiple isolated plants. However, at present, dedicated solutions are applied in different factory [36]–[38]. Only recently, standards specifically designed have been gradually proposed. That means different protocols and semantic data flows within different factories. Therefore, the data of multiple domains cannot directly interact and understand.

Also, identity resolution systems can make equipment better managed. The current research on devices mainly focuses on the operational phase, including how devices behave, operate, communicate and interact with other devices during operation. However, life cycle stages before and after the operation are ignored [34]. The problems to be solved and the supporting technologies in these two stages are different. In the operation phase, devices communicate with others in the same transmission layer. In contrast, to implement device management, the device needs to submit device information to the upper layer, thereby achieving large-scale optimizing. Reference [39] depicts an example of IIoT at the production site of a general factory. Sensor information is collected to the gateway and then managed at the upper layer. As a supporting technology of the framework layer in IIoT, identity resolution systems realize various subject identification, including terminal, network equipment, service resources, data, etc. Through unified identification of various data, multi-system joint optimization and unified resource scheduling are realized.

Besides, an identity resolution system can provide unified methods to upper-layer applications on object identification, addressing and understanding. And it is the basis to meet the homogeneity requirements of data in big data analysis [40]. Besides, the construction of the identity layer can simplify the service provider procedures. If the identity protocol layer is missing in the network, the responsibility for identification and verification will be transferred to the service provider. The service provider acting as the issuer and verifier of the identifier will duplicate information, resulting in low-efficiency [41]. In particular, in IIoT, a large part of service providers are manufacturers. They are not computer and Internet experts, and it is difficult for them to providing complex ICT services. Shielding manufacturing companies from complexity can propagate IIoT better. Therefore, a unified scheme on identification, addressing and reference metadata for the objects in the IIoT is necessary to provide. So that the data can be easy to understand and analyze, facilitating unified optimization between systems.

Typical applications of the identity resolution service includes product life-cycle management [11], [12], supply chain management and traceability [13]–[17], and smart logistics [18]–[20], etc. Product life-cycle management covers requirement analysis, design, manufacture, usage, maintenance, service, reuse, remanufacturing and scrap. It focuses on monitoring real-time state, planning for the whole service process, and coordinating fundamental benefits among product, user and environment [35], [42], [43]. Through supply chain management, tagged products can be traced across the supply chain and business event data can be exchanged among supply chain partners in real-time [44]. Smart logistics can enhance the adjustment to the market changes through resource planning, warehouse management and transportation. It can improve the quality of service, lower the prices of storage and production [45].

We use an example to illustrate the general role of the identity resolution system in IIoT. Product traceability is a typical application of it, as shown in Fig. 3. From warehousing to selling, the production information is stored in different information systems in a distributed and isolated manner. This approach is not conducive to enterprise management and service intelligence. Through the identity resolution system, scattered product information can be correlated. Users can obtain traceability information on the product by only providing a unique identifier.

B. The Design Principles of Identity Resolution Systems for the Industrial Internet of Things

1) Multi-Type Identity Subject Supporting: The types of identification objects in IIoT are various, covering a wide range, including materials, equipment, network elements, services, operators, etc. So multi-type identity subject supporting is required in identity resolution system to meet the diverse and customized needs in IIoT.

2) Compatibility: There are currently multi-standard, multi-protocol, multi-identifier formats coexisting during industrial production, which brings great challenges to retrieving and understanding the data. Even in different departments of the same company, the data standards used are sometimes different. As a result, inconsistent data caused the automatic data analysis to fail to complete [46]. Therefore, heterogeneous resource identity management is becoming more important. The identity resolution systems in the IIoT should be compatible with the existing heterogeneous identification and resolution methods both inside and outside the factory to realize data interoperability. And other identifier formats

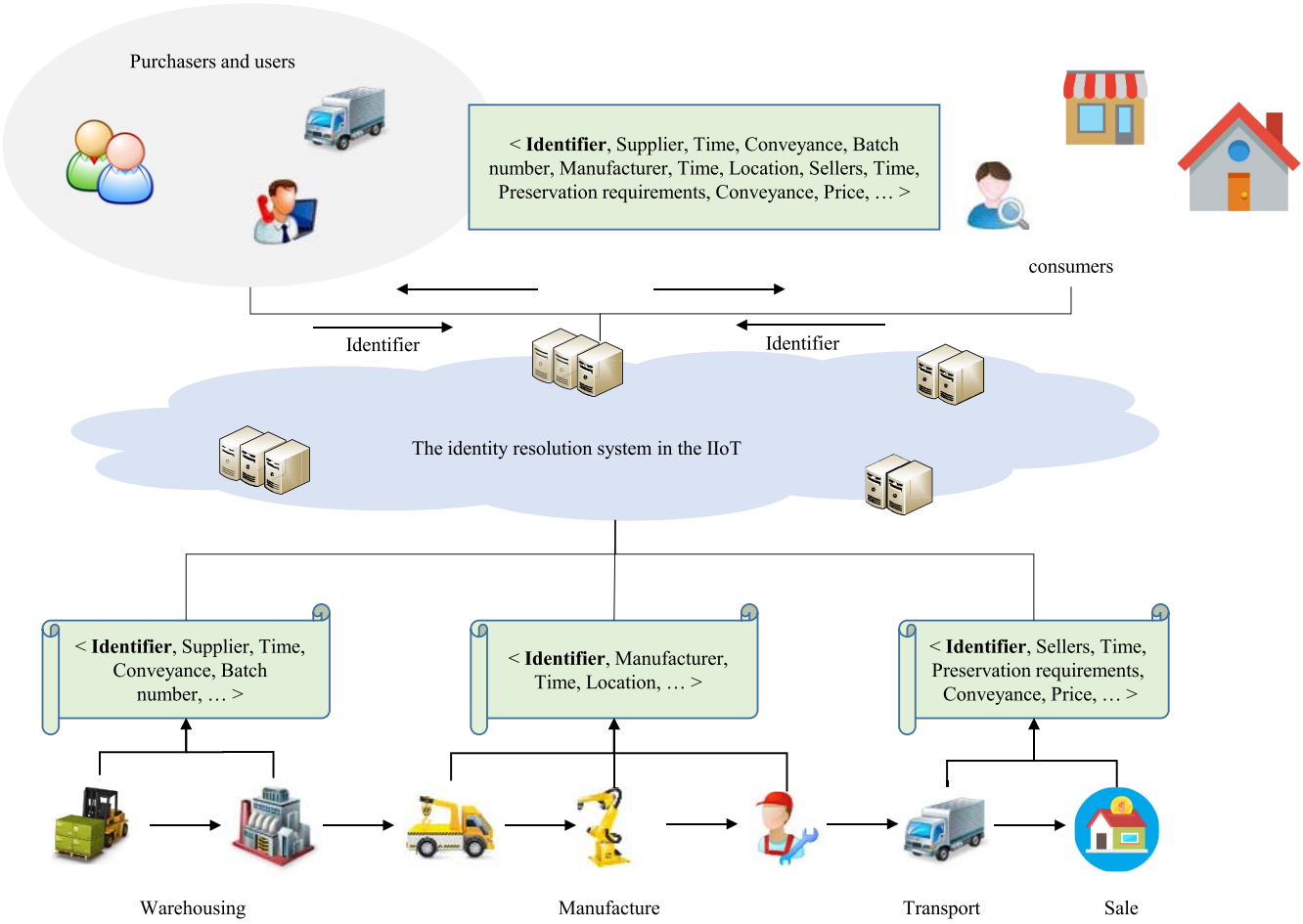


Fig. 3. Application of identity resolution system in product traceability.

and retrieval protocols should be able to join the system seamlessly.

3) Ultra-Low Latency: Industrial production is lower tolerance in latency than IoT [47]. However, identity resolution systems in IIoT face more challenges in achieving ultra-low latency, including 1) multi-identifier format mapping, 2) multi-protocol conversion, 3) difference and customized demands, 4) very high data volume, and 5) high concurrency of the resolving request.

4) Security and Privacy: IIoT connects tens of thousands of assets and is closely related to industrial production and personnel safety. Moreover, the service model of IIoT is complex. The owners of the identification objects in IIoT are intricate and may come from different countries and enterprises. Especially, the ownership of the object will change in real-time based on the outcome of the transactions. Therefore, the identity resolution system in IIoT should ensure security and privacy, including authentication, privacy protection, operation credibility, Anti-DoS/DDoS attacks, and critical business information not be exposed during the resolving process [48]–[50].

5) Fairness: The identity resolution service in IIoT should guarantee fairness, providing neutral and unbiased services for each user. Currently, mainstream identity resolution systems mostly adopt a centralized architecture, and the root domain is controlled by a single entity [51]. This kind of structure has a single point of risk and has raised many concerns with industrial and political communities [52]. With global trade development, corporate cooperation is deepening, and applications such as supply chain management may cross-enterprise or cross-national. Suppose the neutrality of resolution service is ignored. In that case, the service node has the risk of being kidnapped by a special authority, which may cause the service of enterprise unavailability. In IoT, service failure will not bring critical consequences. However, in IIoT, the interruption of processes will impact production and bring potential physical threats [53]. Therefore, it is necessary to design an identity resolution system in IIoT that is peer-to-peer, fair and co-managed by multi-stakeholder.

6) Efficiency: The efficiency of a resolution system has three aspects meaning, including 1) the description is efficient, that is, the description of both the identifier and resolution result of objects is sufficient and less redundant, 2) the architecture is efficient, that is, the architecture is robust with less redundant, 3) the resolution mechanism is effective. On the one hand, the format of the request and response packets is sufficient and concise. On the other hand, the interaction and authentication mechanism in the resolution service should be reliable and brief.

7) Scalability: The identity resolution services of IIoT should be scalable. Its architecture needs to be forward-looking in design and often requires a generic, modular model. A scalable identity resolution system should have the ability to expand to meet the demand in the future massive data scenarios and new identification schemes. The scalability of an identity resolution system should be considered from the following two aspects: 1) scalability at the protocol level. A proper identity resolution system should support the seamless addition of other identity resolution protocol subdomains. 2) scalability at the system level. On the one hand, it is necessary to ensure sufficient identifier spaces to identify future massive digital objects. On the other hand, the addition of service nodes should have little or no impact on existing services.

8) Customized Service Supporting: First of all, different industries often have different requirements for resolution services. For example, transportation has higher requirements for the delay, while forestry is more sensitive to cost. Therefore, the resolution system in IIoT should be able to provide the customized performance of services for different industries. Besides, various applications often have different requirements for resolving results. The resolution results may include URLs, IP addresses, product profiles, logistics information, etc. So the resolution system in IIoT should allow users to customize the type of resolving results.

C. The General Function-Based Reference Framework for Identity Resolution Systems

We summarize the key functions and properties, which constitutes a general reference framework for reviewing an identity resolution system. No matter how an identity resolution system implement, it needs to provide these functions via different technologies. The different technology selection determines the natural pros and cons of systems. And the function-based description framework we proposed can make the deployment of the system in a specific application more flexible. Because it can help researchers understand the functional composition and technical selection of an identity resolution system. It is useful for researchers to make trade-offs based on requirements and design their systems for a specific IIoT application. For example, due to the various problems of traditional tree-shaped identity resolution systems, many researchers build novel systems based on Distributed Hash Table (DHT) [54] or blockchain [55]. In this process, only some functions, such as the coding scheme, are retained. At the same time, the resolution service is replaced by other technologies.

Key functions of an identity resolution system contain the identifier format, identifier allocation mechanism, registration, resolution, data management, security, etc. According to the service requirements, research status and the commonalities of related standards, we focus on the five aspects of the identification, resolving, security and compatibility, as shown in Fig. 4. In this article, we describe and compare systems based on this framework in Section III and Section IV-B.

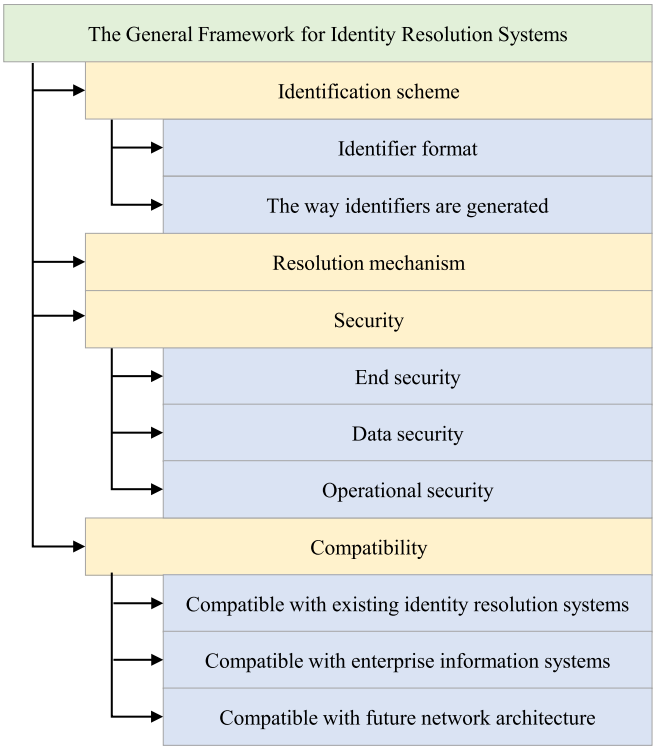


Fig. 4. The general reference framework for identity resolution systems.

1) Identification Scheme: The identification scheme is used to assign and manage unambiguous identifiers of people, goods, materials and industrial equipment involved in industrial production. Thereby supporting awareness of the physical world, information retrieval, and various related applications. An identity scheme consists of two core issues, identifier format and the way identifiers generated.

According to the structure, the existing identity format can be divided into two types: hierarchical identifiers and flat identifiers. Hierarchical identifiers are often cascaded by multiple semantic characters, and it is usually human-friendly but easy to be imitated and forged, such as domain names [56]. Hierarchical identifiers have many advantages, such as supporting multicast by nature and strong scalability. In addition, resolution requests have a long tail effect, that is, most requests are used to access a small number of resources. Hierarchical identifiers can cache the most frequently accessed resources level by level to reduce the burden on service nodes. However, the semantics of hierarchical identifiers limits the survival time of the identifier to a certain extent. For example, if the information of the owner is written in a sub-identifier space, the identifier will be invalidated when the object owner changes. Flat identifiers consist of a series of irregular numbers or strings, usually got by hashing, which are always secure, but difficult for humans to understand. So, it is not conducive to human access to the information behind them. A flat identifier often has a fixed length and can be matched more quickly. Also, flat identifiers support self-authentication and have better stability, compared to the variable-length hierarchical identifier. However, the identifier space of flat identifiers is generally bounded and identifier aggregation is not possible. As a result,

the size of the database is often large, which limits the scalability of the system. Essentially, the scalability problem is open caused by non-aggregatable [57]. Besides, the change of the resource content or the upgrade of the hash function may cause the original identifier to be invalid, thereby affecting the retrieval and query of the resource.

There are two existing identifier generation methods: centralized and decentralized. In a centralized manner, identifiers are assigned and managed by identity service providers, which is highly efficient and easy to manage. However, the problem with centralizing trust has become evident in recent discoveries of mass surveillance and censorship programs as well as information leakage through hacking incidents [58]. On the contrary, there is no authoritative service node in a decentralized identity resolution system, and decentralized identity management is often introduced through public and private keys. Users can generate their own identifiers without registering in a central authority. This method is safe and antitrust. However, it needs to reach consensus across the network so it lacks efficiency. Besides, a decentralized identity resolution system requires a reasonable mechanism to ensure that identifier space resources are not abused and wasted.

2) Resolution Mechanism: A resolution mechanism defines the retrieval process of resource, that is, describes how the system queries corresponding information according to the object identifier given by the user. According to the architecture, the existing identity resolution system can be divided into hierarchical architecture and flat architecture, and the corresponding resolution schemes are hierarchical resolution mode and flat resolution mode, respectively.

Hierarchical architectures often use a tree structure. In a hierarchical identity resolution system, each service node manages a domain, and the resolution service is done recursively or iteratively. This type of structure has many advantages, such as simplicity, scalability, and ease of deployment. However, they also have limitations in robustness and fairness. Because the hierarchical tree structure will face the risk of a single point of failure, which limits the robustness. Besides, the power of each service node is different, the root node has the highest authority, and the parent node has higher authority than the child node. Consequently, the parent node can block all child node services, which leads to a defect in fairness.

In a resolution system with flat architecture, the management rights of each service node are equal. Therefore, this kind of architecture can avoid the resolution services being Kidnapped by other organizations through technical or non-technical means. So, it can facilitate the construction of decentralized, equivalence, and autonomous resolution ecology. The flat structure can be implemented using DHT or blockchain.

DHT is a class of distributed peer-to-peer storage system without a central server. In an identity resolution system built by DHT, each service node constitutes a P2P network, and the entries are stored fairly on each service node through a certain DHT protocol. When the resolving request comes, each service node will forward it through the DHT routing algorithm, until the request reaches the target node. Resolution systems constructed in this way is fairer and more robust. However, this kind of architecture and its corresponding resolution mechanism has limitations in latency. The resolution complexity is logarithmically related to the number of service nodes, and the resolution efficiency is lower than the hierarchical resolution. Moreover, its distributed resolution architecture does not have a central node, which will obstruct data collection, and it is difficult to mine and analyze the resolution data.

Besides, blockchain is a promising technology, which is a distributed and decentralized ledger contains connected blocks of transactions. In the resolution system built by the blockchain, the entries are stored on each blockchain node, providing trustworthy, transparent, tamper-resistant and consistent resolution service by a group of nodes without a central authority [59]–[61]. However, deploying blockchain demands huge computing resources. Meanwhile, nodes should synchronize massive block data and deal with numerous transactions, which consume a lot of storage capacity and bandwidth. So, these kinds of systems have shortcomings in resolution efficiency and processing speed.

3) Security: Security and privacy are important for an identity resolution system, mainly including three aspects, that is, end security, data security, and operational security, and the detail is described as follows. 1) End security includes client security and server security. Client security includes the authentication of the client, privacy protection, and even anonymous request in some specific scenarios. Also, the client attack should be blocked, such as a reflection attack. Server security includes the authentication of the server, anti-cache poisoning, anti-hijacking, service registration abuse, DDoS attacks, etc. 2) Data security includes security of transmission and data storage. It means that massive resolving requests and responses are not stolen and tampered with during the transmission of the public network. The above vision may be achieved with the help of the public key infrastructure by encrypting and generating information digest. On the other hand, in the IIoT scenario, data is often stored in a distributed manner rather than being centrally managed. Data storage security is used to secure data during storage, i.e., data is not exposed, stolen, and tampered with. 3) Operational security means that the operations of the participants are following their permission requirements. On the one hand, the resolving request of clients needs to comply with their permissions. On the other hand, the service provider should add, delete, and modify the resolution entries legally, which may be achieved via various access control techniques.

4) Compatibility: Compatibility is an important attribute of the identity resolution system and determines whether the system can be widely used. It mainly includes three aspects, as follows. 1) Compatible with existing identity resolution systems. There are two available approaches to make two identity resolution systems compatible currently. First, the data of the original system be re-registered in the new system, so that the new system can be compatible with the old system. Based on this way, the service reachability is higher but the resource cost is large. Second, there is no need to re-register and only train a classifier at the entrance to the new system. When a resolving request arrives, the specific system corresponding to the request can be intelligently identified. The advantages of

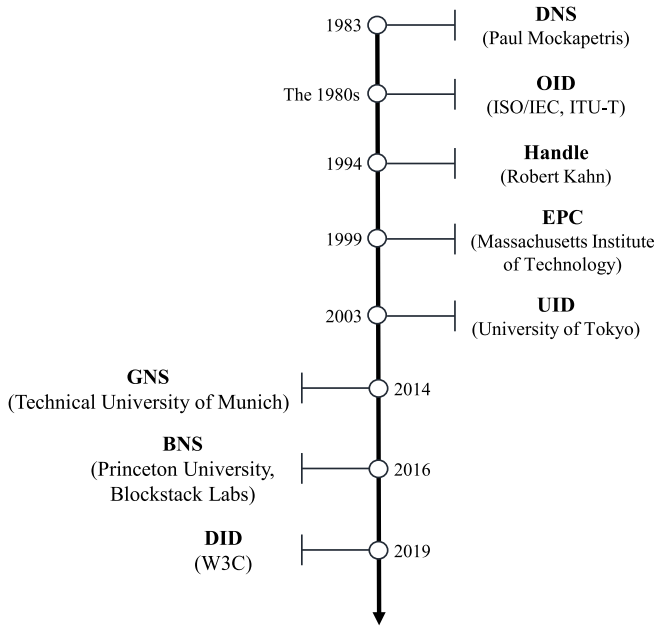


Fig. 5. Timeline of identity resolution milestones. Seminal projects are shown on the left-hand side, while core standards are shown on the right-hand side.

this way are low overhead and multi-identity resolution system interoperability support. But the classifier algorithm design is more difficult. This kind of solution is suitable for scenarios where it is not necessary to construct a unified identifier. 2) Compatible with enterprise information systems. Most companies currently have their digital systems that provide local identity resolution services. Therefore, the solution provided by the identity resolution system for local resolution service access needs to be considered. 3) Compatible with future network architecture. The network architecture may evolve in the future, such as IPv6, Information-Centric Networking (ICN) [62]. So, it is necessary to consider its compatibility with future networks. For example, some researches on the Handle system shows how to deploy it in the Named Data Networking (NDN) [63] and the generation of interest packet and data packet [64]–[66].

D. Classification of Existing Identity Resolution Systems

In Section II-C, we give a general function-based reference framework. This framework describes the functions provided by every identity resolution system. Each system will adopt different implementation and technical routes to achieve the same function. Before introducing the specific functions of these systems, we present the common coarse division of them to help readers have a rough understanding of this field.

There are currently four influential standards and three important projects, as shown in Fig. 5. Specifically, the following have been standardized: Electronic Product Code (EPC), Object Identifier (OID), Handle, and Ubiquitous ID (UID). And the following are important projects: GNU Name System (GNS), Blockstack Naming System (BNS), and Decentralized Identifiers (DID). These systems can be classified by different dimensions: based on design ideas and based on architecture.

1) Classification Based on Design Ideas: According to the design ideas, the existing identity resolution systems can be divided into the following three categories: 1) The designed system is an upper-level application of DNS, providing Identity resolution service for various objects such as goods, services, items, and information, etc., such as EPC, OID, DID. 2) The designed system does not rely on DNS and directly provides identity resolution services based on TCP and UDP, such as Handle and UID. 3) The designed system is a complete alternative to DNS, designed to replace the existing DNS infrastructure. And further, the designed system has expanded the identification subject and resolving results to support identity resolution services for users, organizations, and things, such as GNS and BNS.

We first classified based on whether an identity resolution system is based on DNS, since it is related to the performance and robustness of the system. And whether based on DNS is a very important point for examining an identity resolution system. The details are as follows: 1) It affects the deployment difficulty and cost. 2) The DNS pitfall will be inherited. 3) Building an identity resolution system based on DNS will affect the DNS as feedback.

The implementation difficulty and cost of the system designed based on different design ideas are different. The first type of identity resolution system is the easiest to deploy and does not require changes to existing network infrastructure. However, due to the application of the DNS infrastructure, the service efficiency of such identity resolution systems is relatively low. In addition, services that such systems can provide are limited by DNS, so they are generally stiff. In contrast, the second type of system is more flexible and supports new service design. However, the design of such systems is more difficult and the design cycle is longer. And the third type of system is the most revolutionary, equivalent to redesigning the DNS infrastructure. Such systems can not only solve existing DNS problems but also be suitable for future network needs. However, this system requires network device program updates, which makes its design, implementation, and propagation more difficult.

Building an identity resolution system based on DNS may inherit the defects of DNS. The single point risk of DNS may cause services to be hijacked by special organizations through administrative or technical means. As a result, the resolving service fails, which is intolerable in IIoT, because it is related to production and efficiency. Besides, the single-point structure of DNS is prone to overload, resulting in high service latency and weak resistance to DDoS attacks. On the other hand, DNS transmission in clear text and may face the risk of privacy leakage. Attackers can listen to which identifier the user requests to resolve, which may reveal business secrets.

Moreover, if the identity resolution system is based on DNS, a large amount of traffic in IIoT may be inflowed into DNS. That may affect the operation of DNS. Since DNS is an important infrastructure of the Internet and maintaining the stability of DNS is essential.

2) Classification Based on Architecture: Based on architecture, the existing identity resolution systems can be divided

TABLE II
THE CLASSIFICATION OF SURVEYED SYSTEMS

System	Classification based on design ideas	Classification based on architecture	References
DNS	-	-	-
EPC	• Upper-level application of DNS	Centralized	[67]–[71]
OID	• Upper-level application of DNS	Centralized	[72]–[75]
Handle	• Does not rely on DNS • Directly provides identity resolution services based on TCP and UDP	Centralized	[76]–[78]
UID	• Does not rely on DNS • Directly provides identity resolution services based on TCP and UDP	Centralized	[79]–[83]
GNS	• A complete alternative to DNS, designed to replace the existing DNS infrastructure	Decentralized	[84]–[86]
BNS	• A complete alternative to DNS, designed to replace the existing DNS infrastructure	Decentralized	[87]–[91]
DID	• Upper-level application of DNS	Decentralized	[92]

into the following two categories: 1) The architecture of the designed system is centralized, which means that authoritative servers often exist in the system to be responsible for identification allocation and security. Such systems are EPC, OID, Handle, UID, etc. 2) The architecture of the designed system is fully distributed and decentralized, that is, there is no central authoritative server to ensure security, assigning identifiers to new users, and no authoritative server to guide new users to join the identity resolution service. Such systems are GNS, BNS, DID, etc.

We classify based on the architecture of the system since some applications in IIoT involve the choice of centralized and decentralized systems. The centralized system has low overhead and high efficiency and is suitable for the internal management of an enterprise. A decentralized identity resolution system often contains consensus mechanisms. In IIoT, there are a large number of transactions and services across companies and organizations. They often apply heterogeneous identity systems and desire fair and equal cross-border identity services. The decentralized system can better match the above-mentioned requirements. It supports multiple organizations to negotiate the transmission protocol and data format. Besides, in IIoT, there are rapid growing decentralized distributed heterogeneous devices at the edge, and the identity service between them can also self organize in a distributed identity resolution system.

III. POTENTIAL IDENTITY RESOLUTION SYSTEMS FOR THE INDUSTRIAL INTERNET OF THINGS

In the following sub-sections, we discuss the key functions and properties of a number of identity resolution systems according to the reference framework we present in Section II. And further, give a comprehensive comparison in the next section.

A. Electronic Product Code

1) Overview: Radio Frequency Identification (RFID) is a non-contact automatic identification technology. It uses Radio Frequency (RF) signal to automatic identity stationary or moving objects. The RF tag is generally installed on the product, and the data stored in it can be obtained by the RF reader. Therefore, RFID can be used to track almost all physical objects [67]. So it always used to manage the supply chain of the company to reduce costs.

In 1999, MIT established the Auto-ID Center and proposed the concept of EPC. After that, Cambridge University, Adelaide University, and Fudan University successively participated in the research and development of EPC. Currently, EPCglobal, formed by EAN and UCC, is responsible for the promotion and application of EPC in the world. EPCglobal establishes and maintains an EPC network worldwide to ensure automatic real-time identification of items. It let international trade transparent and visible, improving the efficiency of the global supply chain. EPC system is based on Internet technology and RFID. Each object is assigned a unique code through global unified identification technology. In other words, RFID is an integral part of EPC, and EPC is one of the important applications of RFID. The basic idea of EPC is to use existing computer networks and information resources to store data, thus forming a network pointer with the smallest amount of information.

EPC provides a unique identifier for a physical object, and its system consists of the following three core parts, as shown in Fig. 6. 1) EPC coding. It is the core and key of the EPC and describes the coding scheme for identifying objects. 2) Radio frequency identification system. It is used to automatically acquire EPC. It is connected to the information system and obtains information stored by the EPC tag through wireless sensing. 3) Information network system. It is software support for the EPC system. It first processes the identifier read from the Radio frequency identification system. Then return

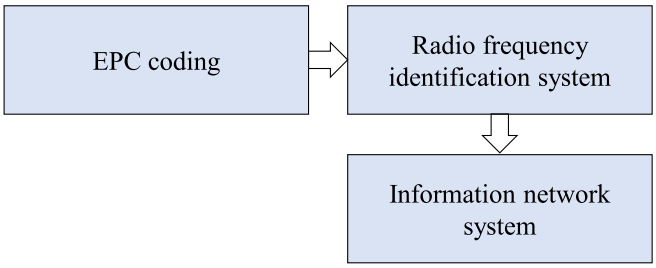


Fig. 6. The architecture of EPC.

the information corresponding to the identifier through the resolving service.

2) Identification Scheme: EPC codes the entity, establishing a global information exchange language via uniform coding. EPC coding is an extension of the original global unified coding system and is compatible with the widely used EAN.UCC coding standard. EPC coding is in all-digital form and is not subject to the local language, economic level, political views, and is a non-discriminatory code.

There are many kinds of EPC, corresponding different formats and lengths. In general, an EPC can be divided into the following four fields. 1) Version number. This field is used to describe the length, type, and encoding structure of the EPC. 2) Domain manager. This field describes the corresponding manufacturer's information. The lengths of the Domain manager of different types EPC are different. 3) Object Class. This field is used to distinguish different product categories. Each product corresponds to a class number in each EPC Domain manager. EPC domain managers are subject to certain constraints when assigning object classes, in addition, it can assign at will. 4) Serial number. This field indicates a specific product within an object class under the corresponding manager field. The storage capacity of EPC tags is relatively small, so binary EPC is stored.

EPC does not contain any product information describing the product name, location, etc. The data stored in the EPC includes the Embedded Information and the Information Reference. Embedded Information includes item weight, size, expiration date, and destination. The Information Reference contains property information about the item. EPC system generates identifiers in a centralized way. And the act of assigning a new EPC and associating it with a particular physical object is called "commissioning".

3) Resolving: EPC resolution service can be seen as a simple lookup service. Through EPC resolution services and network technology, information can be shared among trading partners in the global supply chain. ONS plays an important role in resolving service [69]. ONS relies on DNS. ONS server does not store specific product information, but stores mapping data. These mapping data describe the address of the EPC IS storing detailed information of products. In general, the input of ONS is EPC, and the output of ONS is the address of an EPC IS in the form of a URL. Then the application can access the resolved EPC IS address to obtain detailed information about related items.

Specifically, EPC resolves iteratively, and the EPC resolution system is composed of four parts: TAG Reader, Local System, DNS infrastructure, and EPC IS, as shown in Fig. 7. (1) Tag Reader. The component reads an EPC from the RFID TAG, which is a binary encoding. Then send the read EPC to the Local System. (2) Local System. When an EPC resolving request comes, the Local System first converts the EPC into a standard resource identifier format "EPC URI". Then, EPC URI is passed to ONS Resolver. ONS Resolver converts the EPC URI into a domain name format via a specific conversion method. Finally, ONS Resolver sends the converted EPC domain name to DNS infrastructure via a Naming Authority Pointer (NAPTR) request. When receiving the NAPTR resource record returned by DNS, ONS Resolver will extract the URI of the EPC IS server, and then sends a query request. (3) DNS Infrastructure. This part follows ordinary DNS rules. As mentioned above, ONS is an extended application of DNS. It provides resolving service via the NAPTR record. NAPTR is a type of DNS record that maps between sets of URNs, URLs, and plain domain names and suggests to clients the protocols available for communication with the mapped resource. When the DNS infrastructure receives a domain identity resolving request from the Local System, it returns NAPTR resource records or error messages. (4) EPC IS. EPC IS stores specific product information corresponding to the EPC. It responds to user requests when the query arrives.

In addition, ONS supports two types of resolution services, static query, and dynamic query, as shown in Fig. 8. The resolving results of the static ONS point to a fixed goods supplier. That means a given EPC always points to the same URL. The static ONS service enables related manufacturers in the supply chain to form a serial chain. Therefore, it requires a higher robustness of the system. Under static ONS serial chain, if one server fails, the resolving will fail. The resolving results of the dynamic ONS point to all managed entities in a supply chain. In this way, more detailed information about the entire life cycle of the product can be queried. And the failure of a single node will not affect the entire service, so dynamic ONS is more robust.

4) Security: The provided security mechanism of the EPC system is mainly at the perception layer. And Application layer security depends on the DNS because ONS is the upper layer application of DNS. EPC system mainly enhances its security from four aspects: EPC, EPC tags and readers, discovery services, network information. And the solutions can be roughly divided into two categories: EPC security and access control. EPC security mainly enhances the security of the encoding mechanism. In detail, an EPC is only simply an identification number for a specific object and does not contain extra information. That is, electronic tags cannot convey meaningful information. And all production information related to the EPC must be queried in the EPCglobal network. Access control means consumer access to EPC tags is limited. Users are restricted from accessing information through security technologies such as authentication, interface control, and firewalls. This ensures that only authorized users can access EPC IS information.

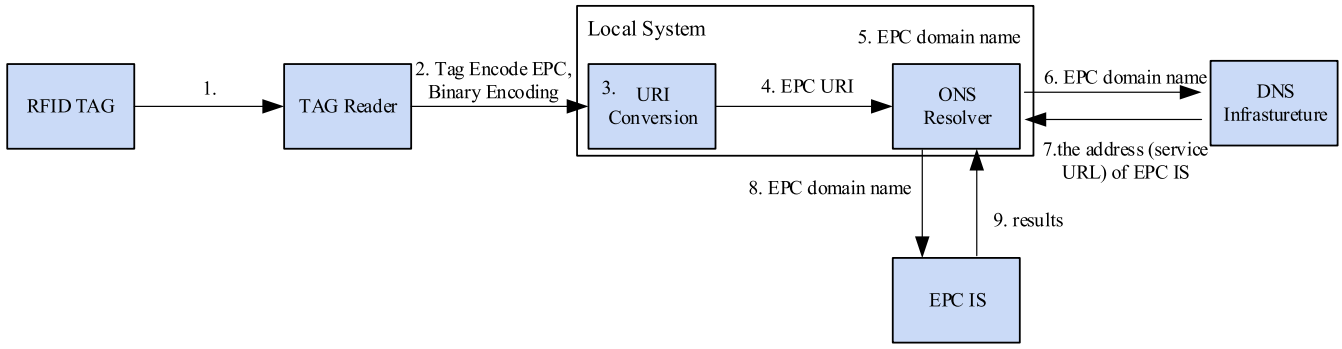
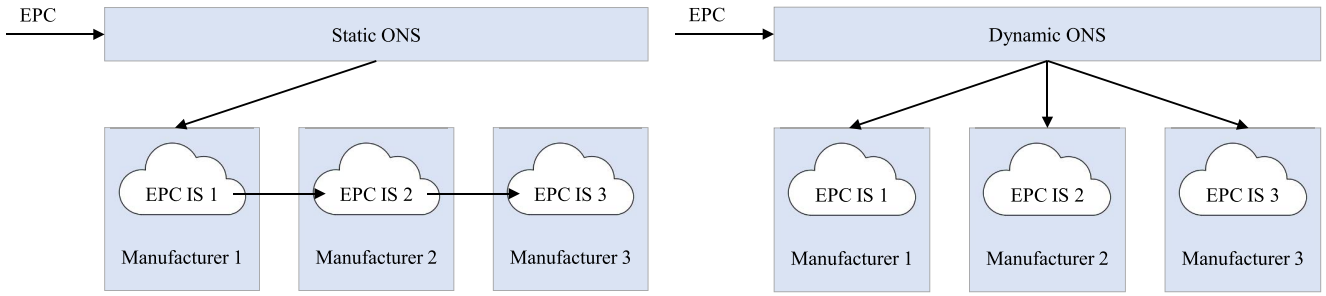


Fig. 7. The architecture of resolution system in EPC.



(a) The static service of ONS

(b) The dynamic service of ONS

Fig. 8. The two kinds of resolution services in EPC.

5) Compatibility: As mentioned above, we can examine the compatibility of an identity resolution system from three dimensions, specifically, the compatibility with other existing identity resolution systems, the compatibility with enterprise local systems, and the compatibility with future network architectures. The EPC system can meet the above three compatibility requirements by writing the URLs or URNs of portal server of the other existing systems, the enterprise local system, and the future identity resolution system in, respectively. However, this is equivalent to re-registering these portal servers in the EPC system, which will bring some additional overhead.

B. Object Identifier

1) Overview: The OID system was jointly proposed by ISO/IEC and ITU-T in the 1980s to identify various objects and services in the IoT. OID adopts a hierarchical tree structure, and the root node is connected to three branches of ITU-T, ISO, and Joint-ISO-ITU-T. An OID identifier consists of a series of numbers and characters, supporting global unambiguous naming of any type of object, including users, network elements, network services, and other physical or logical objects. And once identification, the identifier is valid for life. ISO/IEC and ITU-T standardize the naming, transmission coding, registration process, resolution mechanism, the management system of OID via ISO/ IEC 29168, ISO/IEC 29177, ISO/IEC 9834, ISO/IEC 8824, ISO/IEC 8825, ISO/IEC15962, and ISO/IEC1596 series standards.

The OID system is an upper-level application of DNS, providing OID services by mapping the OID tree as part of the DNS tree. It has many advantages, such as flexible, scalable, readable, and facilitating deployment. Also, the identifier space of the OID system is unbounded, it can support massive industrial data access in the future all over the world. Besides, the OID system supports autonomous management within a domain, and the manager can freely add sub-domains and service nodes. OID is independent of network technology and is not affected by the evolution of underlying equipment. At present, the OID system has been widely adopted in ISO and ITU standards and is currently widely used in electronic medical, information security, network management, finance, food traceability, and logistics, etc. For example, in information security, the OID is used to indicate the hash function, public key algorithm, grouping algorithm, and mode of operation of the X.509 certificate binding. Nowadays, OID has been widely used in RFID, sensors, QR codes, etc. for its advantages of high efficiency, flexibility, compatibility, scalability, and supporting multiple identify objects.

2) Identification Scheme: The OID system identify arbitrary objects such as users, network components, services, tangible assets, and intangible data such as directory structures. The OID adopts a hierarchical identity format, and its coding rule specifies a path from the root node to the identity node. OID provides three commonly used identity formats, namely ASN.1 notation [72], dot notation [73] and OID-International Resource Identifier (OID-IRI), among which point markers and OID-IRI are the most widely used. A comparison of them is shown in Table III.

- ASN.1 notation. Identifiers in this format start with “” and ending with “”. Each sub-identifier space consists of semantic text and numbers, separated by spaces. In detail,

TABLE III
THE COMPARISON OF OID IDENTITY SCHEME

Scheme	Separation notation	Composition of the identifier	Examples	Readability	Efficiency
ASN.1 Notation	Space	Pure number	{2 1}	Hard to read	• Fast retrieval • No redundant information
		Semantic text, all sub-identifier space are attached with number description	{joint-iso-itu-t(2) asn(1)}	Readable	• lack of efficiency • Some redundant information
		Semantic text, some sub-identifier space are attached with number description	{joint-iso-itu-t asn(1)}	Readable	• Performance is between the above two methods • A little redundant information
Dot Notation	Dot	Number	1.3.6.1.6.3	Lack of security	• Fast retrieval • No redundant information
OID-IRI	Slash	Unicode	/Joint-ISO-ITU-T/ASN1	Readable	• lack of efficiency • Some redundant information

there are three ways to implement each sub-identifier space: a) only described by numbers. It close to the machine code. So, the resolution process is faster and safer. But it is difficult for humans to understand and is currently less used. b) described by semantic text, followed by a numeric description. So they are human-readable at the expense of efficiency, having information redundancy in the description. c) described by semantic text or semantic text and numbers joint. In this approach, the numerical description of a few top-level identifier spaces is voluntary, while the numerical description of the lower-level identifier space is mandatory. This kind of identifier considers both human readability and machine retrieval efficiency, and the performance of it is between the above two schemes.

- Dot notation. This format was first introduced by the IETF. Its coding structure is standardized, in which each sub-identifier space consists of only numbers and is separated by dots. Its description is more efficient and there is almost no redundant information. Since consisting of numbers, this kind of identifier is poorly readable, while its retrieval is fast and safe.
- OID-IRI. This format was proposed in the 1990s. Each sub-identifier space of OID-IRI consists of a series of Unicode tags, and separated by slashes. Since consisting of semantic text, its description lack of efficiency to some extent and have a little redundant information. This approach has the advantages of being universal, readable and more flexible, allowing the autonomous definition of identifiers within the domain.

3) Resolving: The OID system is a hierarchical tree structure and resolves in a recursive way [74], [75]. OID resolution system (ORS) provides resolution service and responds to the corresponding mapping data according to the input identifier. At present, the ORS can support the resolution service of dot notation and OID-IRI. Similar to the ONS, ORS also completes resolving operations through the DNS Fully Qualified Domain Name (FQDN) and the NAPTR record. The complete resolution architecture of the OID system consists of four components: application, ORS client, DNS client, and DNS server. The relationship between them is as follows, as shown in Fig. 9. (1) Application. This component is responsible for generating and sending an OID resolving request to the ORS client. The resolving request is consisting of an OID-IRI, an ORS service type, and a security flag. Wherein the ORS service type is a string used to identify the ORS service and is used in the NAPTR resource record. (2) ORS client. This component communicates with the application and the DNS client through the functional interface. It receives the OID resolution request sent by the application and the NAPTR resource record returned by the DNS client. ORS client has two main functions: a) When receiving the resolving request from the application, this component converts the OID-IRI into an FQDN. And then it sends a DNS resolving request to DNS client to obtain the NAPTR record corresponding to the FQDN. b) When receiving the NAPTR resource records,

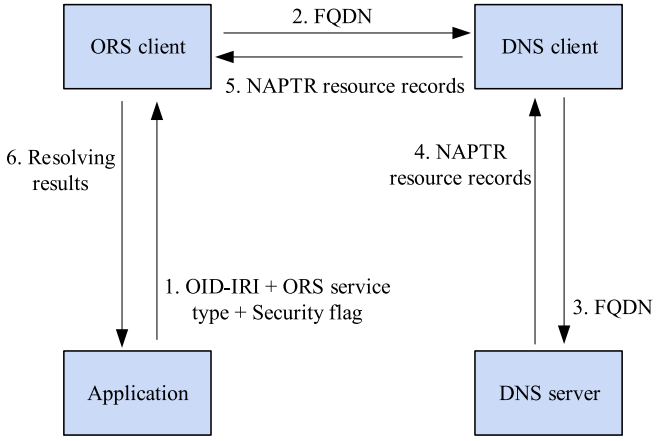


Fig. 9. The architecture of resolution system in OID.

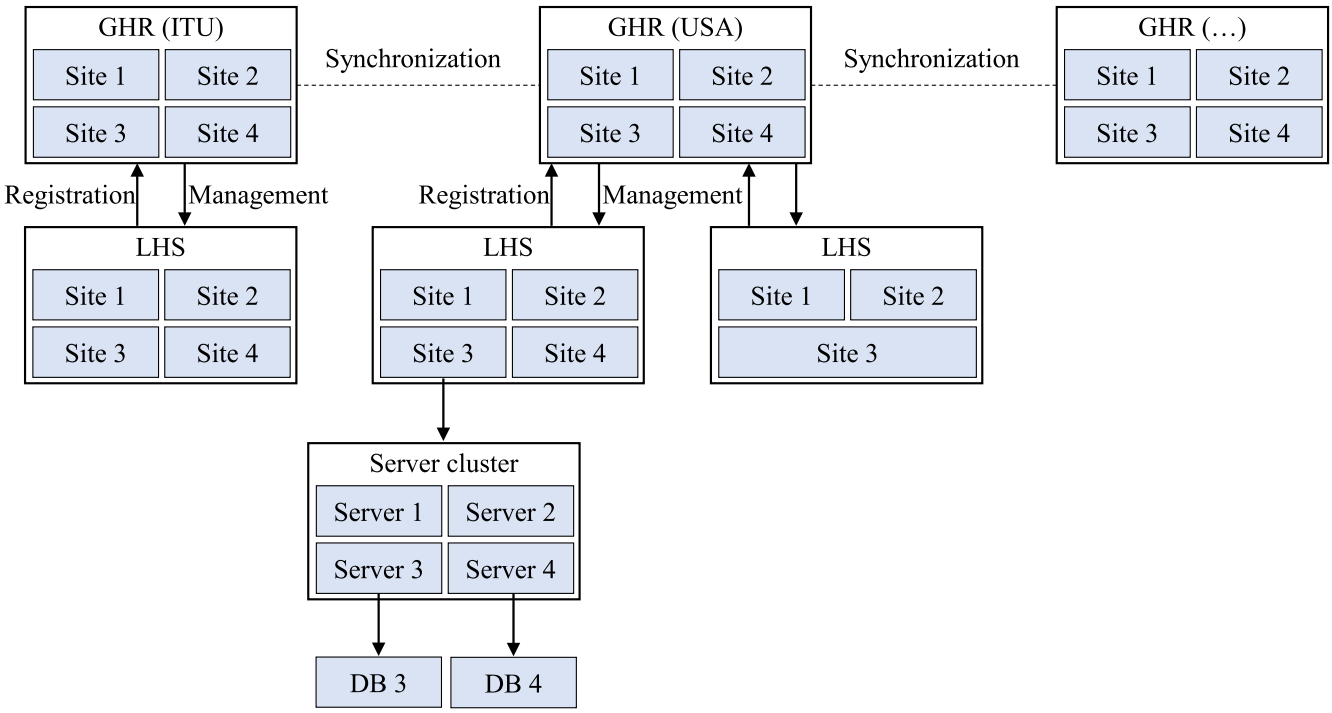


Fig. 10. The architecture of Handle.

the ORS client processes the NAPTR record. And then it returns zero or more information and DNS response code to the application. (3) DNS client. This component is responsible for receiving the DNS resolving request sent by the ORS client. It will forward the request to the DNS server to obtain the NAPTR resource record corresponding to the FQDN. (4) DNS server. This component responds to the request of the DNS client and returning the NAPTR resource record or error message.

4) Security: The security of the OID system mainly relies on the security mechanism of DNS. When a resolving request is generated, the application can decide whether to use Domain Name System Security Extensions (DNSSEC) [93] by setting the security flag. In addition to this, OID does not provide other new security mechanisms. DNSSEC is a series of DNS authentication mechanisms provided by IETF. It supports information digestion and digital signature through hash operation and key encryption. With its help, the information source authentication and information integrity verification can be provided. When the security flag is 1, the OID resolution process supports DNSSEC and requires the DNS server to sign the returned NAPTR resource record.

In short, the ORS does not propose an additional security mechanism, only allowing users to voluntarily use DNSSEC. Because the identity of the resolution service provider can be verified by the digital signature in DNSSEC, the security of the server can be guaranteed. Also, the information digest can be used to verify the correctness and completeness of the data. However, the granularity of client permission division is rough, so it is not suitable for the resolving of industrial critical data.

5) Compatibility: The OID system can compatibility with other existing identity resolution systems, enterprise local systems respectively by writing in the URLs or URNs of their portal servers. In other words, it is necessary to re-register their resolution servers or system portal servers in the OID system, causing some overhead. However, the solutions to provide compatibility with future network architectures are ignored, and there are few kinds of research to solve this problem.

C. Handle

1) Overview: Handle is a global distributed universal identification service system, proposed by Robert Kahn in 1994. It is presented to provide efficient, scalable, and secure global identity resolution services. The Handle system is joined Next Generation Network Research in 2005 and became an integral part of the digital object registry in the GENI project. It is currently operated, managed, maintained and coordinated by the DONA Foundation. The Handle system is the earliest and most used global digital object unique identify system, providing a identifier-to-value binding service. The identifier in the Handle system is also called Handle and can be used to identify any digital object, such as information, services, and other network resources. The Handle system includes a set of reference implementations for identifier spaces and open protocols. Furthermore, it defines encoding rules, the data model, the service model, the operation model, and a globally distributed management architecture.

The Handle system does not rely on DNS and directly provides identity resolution services based on TCP and UDP. It uses a hierarchical service model with multiple root nodes. The top layer of the Handle consists of a number of parallel Global Handle Registry (GHRs). The GHRs are equal and the data is synchronized. And the lower layer consists of Local Handle Service (LHS), as shown in Fig. 10. In the Handle

system, GHR is isomorphic to LHS. Both of them consist of one or more parallel service sites, each of which is a replica of other sites. Also, each service site is composed of different numbers of Handle servers, and all Handle requests are eventually directed to the Handle server. The main difference between GHR and LHS is their services offered. The GHR is responsible for managing services, assigning prefixes, and authorizing identifier spaces. While the LHS is responsible for defining and maintaining the local identifier space. And the prefix must be registered at the GHR. In a typical Handle resolution process, clients can find the target service for each Handle by querying the GHR.

The Handle system is designed for universal identification services. It can serve a large number of entities and allow distributed management over public networks. Also, because its top-level services are equal and support the user-defined identity method, it is more suitable for industrial scenarios. Besides, Handle has characteristics of unique, persisting, multiple instances, multiple attributes, extensible identifier space and efficient distributed service model. It is widely used in digital libraries nowadays and has attracted increasing attention from academia, industry, and government.

2) Identification Scheme: The identifier in the Handle system is called Handle. It adopts a hierarchical format, and each Handle consists of two parts: a prefix and a suffix. The prefix is its naming authority and the suffix describes a unique local name under the naming authority. These two parts are separated by the ASCII character "/": <Handle> := <Handle Naming Authority> "/" <Handle Local Name>

Handle Naming Authority, the prefix, is the creator and manager of Handle. It consists of multiple non-empty naming authority segments, each of which separated by the ASCII character ".", forming a hierarchical tree structure. Handle Local Name, the suffix, is defined by the naming authority and should be unique within its local namespace. A Handle can be globally unique in the system by its unique naming authority and unique local name. For example, suppose a Handle 20.500.12357 */BUPT_FNL*, the prefix of it is 20.500.12357, that is, the naming authority is 20.500.12357. Moreover, its suffix, that is, the local name, is *BUPT_FNL*. It is unique in the Handle system.

The global identifier space of Handle can be thought of as a superset of multiple local identifier spaces. Each local identifier space has a unique prefix, and any local identifier space can be added to the global identifier space by register a prefix. Note that after adding to the Handle system, the binding relationship between the local identifier and the value will not change. In other words, a global identifier can be created by simply using the combination of the local name and a prefix. This feature of the Handle system is very important. On the one hand, it helps to facilitate enterprises to add their own information systems to Handle. On the other hand, it can also make the Handle system more compatible with other identification schemes.

3) Resolving: The Handle system provides binding services between identifiers and values, and each Handle can be resolved into a set of values. The type of each value is predefined, and the value can be an item description, a message digest, a URL, or other customized information. The Handle system adopts a hierarchical resolution architecture and an iterative resolution method. The Handle system consists of two layers: GHR and LHS. And the complete resolution architecture consists of three parts: Handle client, GHR and LHS, as shown in Fig. 11. (1) Handle client. In the resolving phase, this component first sends the prefix of the Handle to the GHR to obtain the LHS service site information to which the prefix belongs. After that, the complete Handle is sent to the corresponding LHS service site to obtain the resolving results. (2) GHR. This component receives and responds to requests for resolving the prefix of Handle sent by the Handle client. The GHR retrieves the target LHS service site by querying the registration information. Once retrieved, the service site information will be returned to the Handle client. (3) LHS. This component receives and responds to requests for resolving Handle sent by the Handle client. The LHS retrieves the set of values corresponding to the identifier by querying the local database. Once retrieved, the results will be returned to the Handle client.

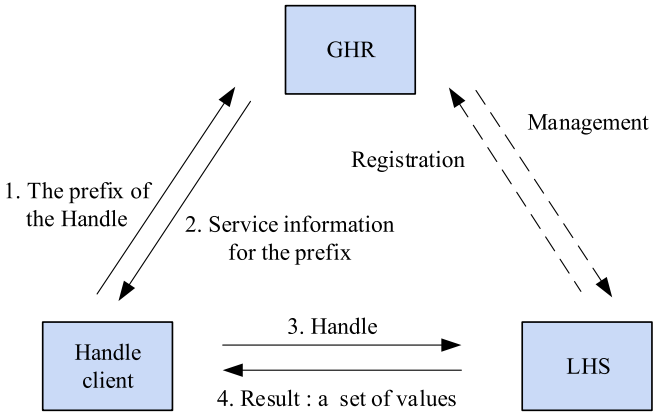


Fig. 11. The architecture of resolution system in Handle.

In addition, to improve the resolving performance, the Handle client can cache the LHS service site information returned by the GHR and use it for subsequent queries. Then, based on the cached information, the client can directly send the resolving request to the corresponding LHS service site without asking the GHR. Also, unlike the single root structure of DNS, the top level of Handle is parallelized. Therefore, the problems caused by centralized DNS management can be partially alleviated. Besides, Handle allows registered LHS to customize its local namespace and resolution mechanism. Therefore, the Handle system supports seamlessly adding other protocol subdomains, and can compatibility with other identity resolution systems.

4) Security: The Handle system is implemented directly based on IP protocol and does not rely on DNS services, meaning that the DNS security mechanism cannot be used. In order to ensure the security of the service, the Handle system has designed a series of security mechanisms, and its main work includes the following three parts.

(1) Administrator and privileges design. In the Handle system, every value must have a data type specified in its

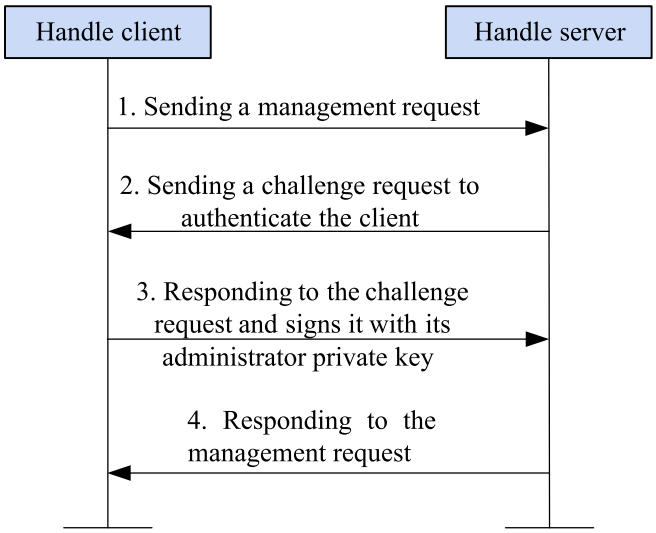


Fig. 12. The challenge-response protocol flow in Handle.

type field. *HS_ADMIN* is a pre-defined Handle data type. And users can set up one or more administrators for each Handle by defining in terms of *HS_ADMIN* values. Any administrative operation (e.g., add, delete or modify Handle values) can only be performed by the Handle administrator with adequate privilege. In the Handle system, authentication of the administrator is required before responding to any Handle management request.

(2) The security of the Handle Client. The client can initiate two types of requests, resolving requests and managing requests. A client needs to be authenticated before responding to both types of requests. If clients initiate resolving requests, the Handle server performs different resolution responses to clients according to their different privileges. If a client initiates a management request, the Handle system authenticates the client via a challenge-response protocol. The challenge-response protocol flow is shown in Fig 12, and Its specific steps are as follows. First, the client sends a management request to the Handle server. Next, the server sends a challenge request to authenticate the client. Then, the client responds to the challenge-response and signs it with its administrator private key. Finally, the Handle server can authenticate the client by verifying the digital signature based on the public key of the administrator. If the verification fails, the client is notified. Otherwise, the Handle server will further check if the administrator has adequate privileges. If so, the server performs the management operation and reports success to the client. Otherwise, a rejection message is returned.

(3) The security of the Handle server. The client can require the Handle server to sign the response with a private key to authenticate the Handle server.

In addition, the Handle system provides distributed data management capabilities, supporting distributed, centralized, cloud storage and other storage methods. So, the Handle system has a stronger content protection mechanism and anti-attack capability than DNS. Also, the Handle system defines authentication mechanisms that support autonomous management of data, access privileges, and user identities. So, it is secure and reliable.

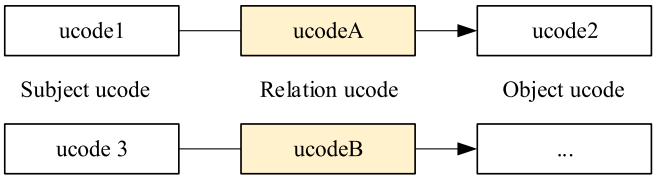


Fig. 13. The ucR unit in UID.

5) Compatibility: In a sense, Handle system compatibility is more prominent in these surveyed systems. Because any local identity resolution system can be added to the Handle system by registering a prefix. This means that the Handle system is not only compatible with other existing identity resolution systems, but also can seamlessly add local enterprise information systems. In addition, the Handle system can compatible with other systems with minimal overhead. The system to be added only needs to register a new prefix and store in the GHR, and does not bring about large-scale data migration. Moreover, there have been some academic studies focusing on how to deploy Handle in the future network. For example, some researches focus on how to wrap resolution requests in Handle into an NDN interest package, and how to wrap the resolution response in Handle into an NDN packet [64]–[66]. So that the Handle system can still be used normally in the NDN network. Due to the real impact on whether a system can be promoted is often its compatibility and deployment difficulty. So, from a compatibility point of view, the Handle system will be a promising identity resolution system, meeting the three requirements we mentioned above.

D. Ubiquitous ID

1) Overview: UID is a context-awareness technology for ubiquitous computing, describing objects and the relationship between objects. The UID Center was established at the University of Tokyo in 2003 and received strong support from the Japanese government and companies. The UID standards are based on the extensive application of The Real-time Operating system Nucleus (TRON). To date, more than 500 companies and organizations all over the world have participated in the release of the industry open standards specification for UID standards and ubiquitous computing systems. UID identifies physical or logical objects such as spaces, addresses, concepts, etc. by ubiquitous code (ucode). And UID establishes associations between ucodes via the ucode Relation model (ucR model). The main body of the ucR model is ucR Unit, and Fig. 13 shows the structure of the ucR Unit. Each ucR Unit consists of three parts: subject ucode, relation ucode, and object ucode. The ucR Unit is used to indicate the relationship between two ucodes. To describe multi-entity and complex context information, UID further splicing multiple ucR units into ucR Graph, as shown in Fig. 14.

Ucode can realize the identification and communication of any object. It is an important technology to realize the ubiquitous computing, IoT and M2M computing paradigms.

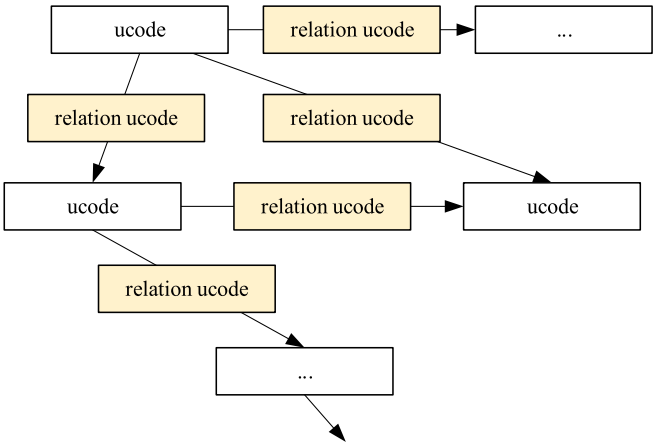


Fig. 14. The ucR Graph in UID.

The ucode related technology was written into the ITU-T international standard in October 2012. Currently, a series of ucode-based Recommendations are being accelerated. They are used to provide object identification, location description, context understanding, and interaction between objects across applications or across organizations. So that the optimal control is performed automatically without manual intervention. Ucode has a good application base and can describe the context, suitable for IIoT. It is expected to be used in applications such as building management, food and medical product traceability, factory facility disposal, tourism information services, and public asset management.

2) Identification Scheme: Ucode is a hierarchical, fixed-length format consisting of a series of human unreadable numbers. Ucode has a basic length of 128 bits and supports length extension. And the length of ucode can be extended to an integer multiple of 128, such as 256-bit, 384-bit, 512-bit, and so on. The ucode identifier space is managed in a hierarchical structure consisting of two layers, a top-level domain, and a second-level domain. Each ucode consists of five fields: version, top-level domain code (TLDc), class code(cc), second-level domain code (SLDc), and identification code (ic). Fig. 15 shows the structure of a 128-bit basic width ucode. The version is used to indicate the version of ucode. The TLDc is used to represent the top-level domain manager of the ucode. The cc field describe whether the ucode has extended the base length and the boundary between SLDc and ic. SLDc is used to indicate the secondary domain manager of the ucode, which is assigned by the top-level domain manager. Different types of ucode have different SLDc lengths, and its specific length is determined by the cc. The ic field is responsible for uniquely identifying the object. There are also many types of ic, and the specific type is described by cc.

Ucode supports the identity of multiple types of objects. In addition to identifying physical entities, ucode can also identify abstract objects, such as concepts, locations, relationships, etc., that can meet the diverse needs of industrial scenarios. Moreover, ucode is fixed-length and made up of a series of human unreadable numbers, so the matching of ucode is fast.

Version	TLD Code (TLDc)	Class Code (cc)	SLD Code (sLDc)	Identification Code(ic)
4 bit	4 bit	4 bit	Multiple types	Multiple types

Fig. 15. The ucode (128-bit basic width) structure in UID [83].

However, the namespace of ucode is limited, and it is difficult to meet the needs of massive data identification.

3) Resolving: As mentioned above, ucR Graph describes the relationship between multiple objects and is stored in the ucode relation database. The ucode resolution system receives and responds to ucode resolving requests. The ucode resolution system retrieves the corresponding ucR Graph in the ucode relation database according to the identifier to implement context-awareness.

Ucode adopts a recursive resolution way. The ucode resolution system consists of four core components: ucR Database Front-End (UDF), ucR Database Node (UDN), ucR Vocabulary Engine (UVE) and ucode Information Service (ucodeIS), as shown in Fig. 16. (1) UDF. This component is deployed within the ucode infrastructure system and is responsible for receiving and responding to ucode resolution requests. After receiving the resolving request, the UDF requests the relevant ucR Unit from the distributed UDN. Then it builds ucR Graph based on these received ucR Units. Then, this component describes the context corresponding to the ucode based on the UVE. (2) UDN. This component is deployed in the ucode infrastructure system and is an independent node in the ucode relation database. It is responsible for participating in the distributed storage of the ucR unit. (3) UVE. This component is deployed within the application. Note that different applications have different UVEs. This component is responsible for providing semantic understanding and search logic for the ucR Graph generated by UDF. For example, extracting location information from ucR Graph is an application-specific UVE. (4) ucodeIS. The component is deployed within a specific application and serves users based on the search results of ucode.

Unlike other systems, the resolving results of ucode is all relevant context information. The application then filters out the required content in the returned context information based on its specific needs and search logic. So, in UID, the resolution results and the object description are more comprehensive. However, the system needs to collect ucR Units from multiple distributed nodes during the resolving process, so the resolution is a lack of efficiency.

4) Security: To meet the security differentiation requirements of different applications, the UID system divides security functions from low to high into seven levels based on the degree of security and privacy protection. In detail, they are data corruption detection, anti-physical replication and forgery, access control, tamper resistance, secure communication with unknown nodes, support for time-based resource management, and support for updates to internal program and security information. Specifically, the UID system ensures data security by constructing data corruption detection,

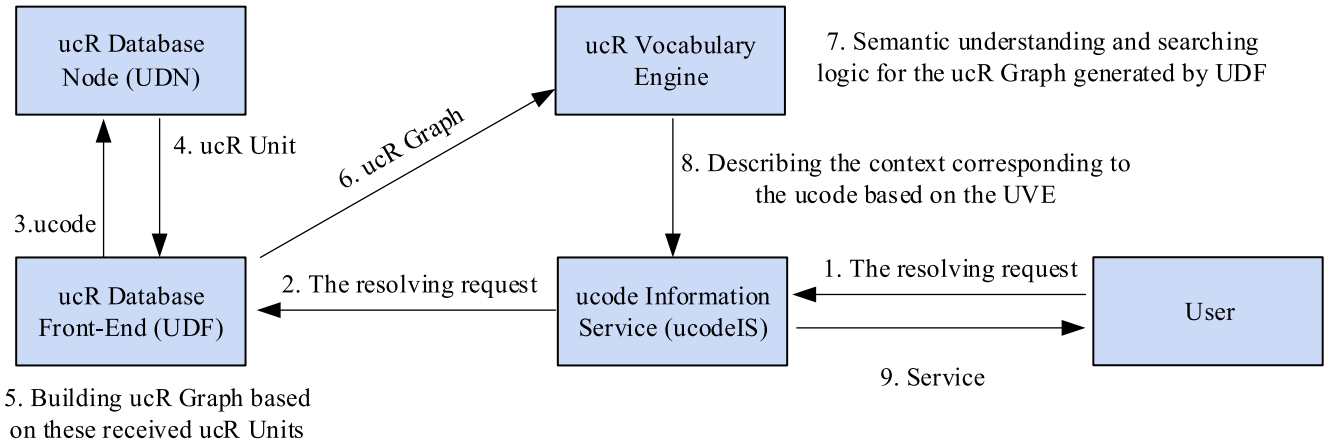


Fig. 16. The architecture of resolution system in UID.

anti-physical replication and forgery, and secure communication with unknown nodes. Ensure operational security by designing access control, supporting time-based resource management, etc. By designing the above seven security functions, the UID system provides fine-grained and flexible security solutions for applications, which can meet the differentiated security requirements of different users.

5) Compatibility: Ucode does not provide a solution compatible with other identity resolution systems. In addition, because the identifier length of the ucode is fixed, ucode is not suitable as a metasystem that implements multiple heterogeneous identifiers interoperability.

E. GNU Name System

1) Overview: GNUnet is a software framework for decentralized, peer-to-peer networking. It developed by GNUnet eingetragener Verein (GNUnet e.V.), and its initial release is published on November 5, 2001. GNUnet offers link encryption, peer discovery, resource allocation, communication over many types of transport, etc. The basic technology of GNUnet is DHT. In DHT, data can be stored randomly and fairly on multiple nodes via a certain protocol. GNUnet is a new network protocol stack for building secure, distributed, and privacy-preserving applications. It is typically run as an overlay network on top of the existing Internet infrastructure forming the basis of a hybrid peer-to-peer mesh and relay backbone for applications to run on.

GNS was proposed for the GNU project by the Technical University of Munich in 2014 and is still being improved. It can support distributed, censorship-resistant, privacy-enhancing identity management, replacing the DNS and X.509 certificate authorities (CAs) with a more privacy-friendly but equally usable protocol. In GNS, each user manages their own zones by a general solution without the need for a centralized service provider. And the control of subdomains can be delegated to zones managed by other users. Moreover, the lookups of records are performed using the DHT algorithm. For interoperability with DNS, domain names in GNS use the pseudo-TLD ". gnu". ". gnu" refers to the GNS master zones and is also the starting point of the resolving process.

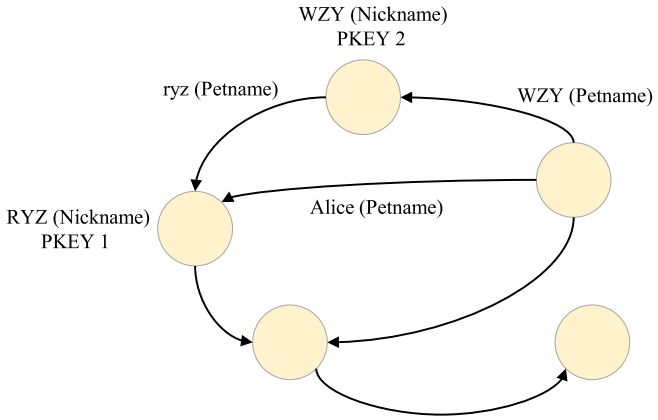


Fig. 17. The architecture of GNS with directed graph relationships.

Instead of insisting on a zone tree, GNS relaxes the relationship between zones to that of a directed graph as shown in Fig. 17. And that is the main difference between DNS and GNS. The GNS has three core designs as described below. 1) GNS is a petname system. Each user can choose a nickname to declare themselves. Meanwhile, each user can select a petname as a label to indicate a new acquaintance. This petname is set to the nickname of the object by default. Users can also assign a petname according to their preferences instead of using the suggested nickname. 2) Drawing on the idea of SDSI/SPKI, GNS allows users to delegate control of subdomains to other users. 3) Support privacy-preserving and integrity protection. In GNS, key-value mapping, queries, and responses are all encrypted.

Since GNS is intended to coexist with DNS, most DNS resource records are used with identical semantics and binary format in GNS. Besides, GNS defines various additional records to support GNS-specific operations. Among them, there are three types of newly defined records worth mentioning: PEKY, NICK, GNS2DNS. Their motivations, characteristics, and indicator number are shown in Table IV. PKEY records are used to implement secure domain delegation in GNS and map petname to keys. PEKY records can be understood as a secure variant of NS records in DNS. The difference

TABLE IV
THE RECORD OF GNS

Type	Motivation	Comparison with DNS	Data entity	Indicator Number
PKEY	GNS delegation: Securely delegate control over a subdomain to another zone.	A secure variant of NS records.	• Public Key	65536
NICK	To publish an indication on what label a zone prefers to be referred to.	Innovation in GNS, no similar record in DNS.	• Nickname	65537
GNS2DNS	DNS delegation: delegate resolution for a subdomain from GNS to DNS.	Similar to NS records.	• DNS Name • DNS Server Name	65540

TABLE V
THE IDENTITY SCHEME OF GNS

Item		Description	Characteristics	The method of generation or publish	Motivation
Public Key		Principals defining a local identifier space	• Global • Securely, Collision-free • Unmemorable	Via publishing PKEY Record	Addressing
Name	Nickname	• No need to be globally unique, only not already be in use within his social group. • Has a one-to-many mapping to keys	• Global • Leak of security, Maybe Collision • Memorable	Via publishing NICK Record	Make it easy for human beings to manipulate keys
	Petname	• Only valid in the local identifier space defined by the key • Has a bidirectional one-to-one mapping to a key	• Local • Securely, Collision-free • Memorable	Generated by users according to their preferences	Make it easy for human beings to manipulate keys

is that the tree structure of DNS is replaced with a directed graph through PKEY records in GNS. NICK records are used by zone administrators to publish an indication on what label this zone prefers to be referred to. This is a suggestion to other zones what petname to use when creating a PKEY record. GNS2DNS records are used to delegate service to DNS. The data entity of them contains a DNS name followed by a DNS server to use. GNS2DNS records are also similar to NS records in DNS.

2) Identification Scheme: As mentioned above, GNS is a petname system. A petname system uses three interrelated types of identifiers at the same time to get three desirable properties in one identity scheme, that is, memorable, global and securely collision-free. These three types of identifiers are: key, nickname, and petname, as shown in Table V. Public Keys are principals defining a local identifier space. And identifiers are only valid in the local identifier space defined by that key. The public key is used for addressing, while identifiers make it easy for human beings to manipulate keys. In GNS, each user manages a zone. Each zone corresponds to a key pair, a nickname, and multiple petnames. The public key indicates the manager of the zone. petname is used for users to instruct other domain administrators in their operation view, which can be arbitrarily specified according to the user's own preferences. Petname only needs to be unique within the local identifier space. The nickname allows the zone administrator to provide suggestions for other users to generate petname, which is published through the NICK record. The nickname does not need to be globally unique, only not already be in use within his social group.

The GNS identification scheme is very clever. GNS implements identity resolution services by combining the advantages of public key, nickname, and petname. Public keys are used in public network addressing, which are operated only by computers, and do not need to be readable by humans. Meanwhile, the petname is used in the user interface. It is not only readable during human operation, but also securely collision-free due to its local characteristics. Also, because users or businesses have an incentive to let others remember their services, they have an incentive to publish nicknames to provide suggestions for other users to generate petnames. In addition, GNS also designed a series of related resource records to complete the mapping between names and public keys to ensure the effective work of the above mechanism. The GNS identity scheme is wonderful, bypassing the necessary trade-offs that the three characteristics cannot meet at the same time as pointed out by the Zooko triangle. It can be both Securely collision-free, global, and human-readable if needed. Besides, on the one hand, GNS can coexist with DNS, on the other hand, the function of GNS is not limited to traditional DNS. In addition to identifying hosts like DNS, GNS can also identify users, things, and organizations. This feature makes its application in IIoT possible.

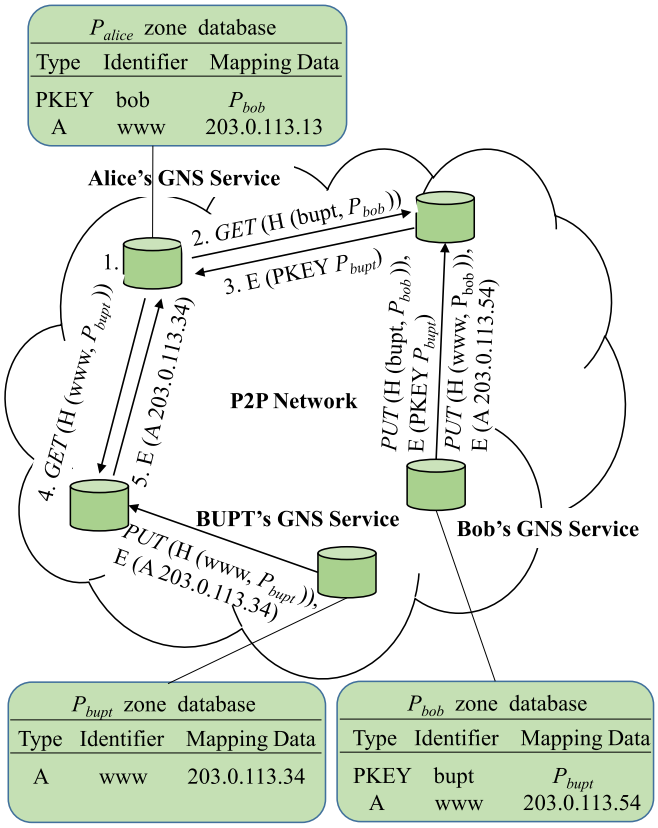


Fig. 18. The resolution process in GNS.

3) Resolving: GNS is an alternative to DNS that maps domain names to IP addresses. Similar to DNS, the GNS resolution is performed sequentially through zone delegation. And as long as the remainder of the identifier is not empty, the resolving continues recursively with the remainder of the identifier. The difference is that, in GNS, the domain name in each user view is different, which is related to the petname they set. So, the key to GNS resolution is the identifier switch. In addition, since GNS is applied to GNUnet, naturally, GNS resource records are published through the DHT algorithm. For example, user *Bob* in the network knows the public key of *BUPT* in real life. In order to facilitate his own operation, *Bob* gave *BUPT* a petname: *bupt*. Then the mapping relationship between *bupt* and its public key is stored in *Bob*'s zone database. In addition, in order to let other users in the network know the mapping relationship, *Bob* publishes *bupt* and its corresponding public key via DHT by the PKEY record. Note that the record is encrypted. Besides, in order for others to access its webserver, *BUPT* publishes its public key and corresponding IP address through DHT by A record. Note that this record is also encrypted. Then we will exemplify the resolving process of GNS below, as shown in Fig. 18. Now, suppose *Alice* wants to access *BUPT*'s webserver using the identifier "*www.bupt.bob.gnu*". Note that, every identifier in GNS uses the pseudo-TLD ".gnu", and ".gnu" is the starting point of the resolving process. The resolution of GNS is performed in an iterative way. The requester will continuously query the mapping data of each field in the P2P network. The detailed steps are as follows. (1) First of all, *Alice* resolves the "*.bob*" field. Since the GNS identifier actually indicates the zone delegation path, *Alice* knows that *bob* has information about the *bupt*. And according to her zone database, *Alice* can know the public key corresponding to *bob*. (2) Then resolving the "*.bupt*" field. After getting *bob*'s public key P_{bob}, *Alice* will query the public key of *bupt* in the P2P network. (3) *Alice* gets the public key corresponding to *bupt* through the pre-published PKEY record from *Bob*. (4)vThen, *Alice* resolves the "*.www*" field by initiating a query request to the P2P network. (5) Finally, *Alice* resolves out "*.www*" through the pre-published A record to obtain the IP address corresponding to "*www.bupt.bob.gnu*".

4) Security: This part will consider the security of GNS from the following aspects. Firstly, From the perspective of system-level, because GNS is stored through the DHT algorithm, all GNS queries go to the same, fully distributed global infrastructure, rather than operator-specific servers. So GNS can avoid attacks on specific servers, both technical and political. Secondly, data security can also be guaranteed, that is, records are safe during transmission and storage. Although GNS public records are stored globally in a distributed manner, they are cryptographically signed, so the privacy of users is still guaranteed. Then, since both queries and replies are signed, end security can also be provided. Finally, operational security is also guaranteed. In GNS, its resource records are encrypted, meaning that only people with permissions and keys can get query results. Also, since the key-value mappings, queries, and responses in GNS are all cryptographically signed, GNS can also provide integrity protection.

Besides, GNS provides a zone revocation mechanism. If a zone's private key gets lost or compromised, the key can be revoked. Different from the traditional explicit query to check whether a key has been revoked, GNS takes advantage of P2P networks to spread revocation information through flooding. Meanwhile, in order to prevent malicious nodes from using this mechanism to initiate denial of service (DoS) attacks, GNS requires that revocations include revocation-specific proof of work. That is, the initiator of revocations consumes expensive computing resources.

5) Compatibility: First of all, GNS achieves DNS compatibility and interoperability by using the pseudo-TLD ".gnu" and partially using the format of DNS records. In other words, GNS is a complete alternative to DNS. Therefore, the deployment of GNS does not affect the normal use of DNS-based identity resolution systems such as EPC and OID. From the current evolution path of GNS, the best scene in which GNS is applied to IIoT is the scenario where the distributed underlying network is required to adopt DHT. Finally, GNS is a natural design under the characteristics of GNUnet's fully distributed networking. So, the most suitable scenario for GNS is a complete DHT network, and it is not suitable for a universal network architecture with a central node. Therefore, GNS has weak compatibility with future network architectures such as ICN and SDN.

F. Blockstack Naming System

1) Overview: BNS is a radical re-design distributed naming system based on Blockstack, which is jointly proposed

by Princeton University and Blockstack Lab. It is designed to replace the centralized DNS. BNS can solve the storage limitation and security problems of Namecoin [87], which is the first system to build a decentralized naming system using a blockchain. The core idea of BNS is to minimize the degree to which users need to trust in a single party, like a DNS root server or a root certificate authority. BNS builds a new decentralized PKI system on the blockchain and enables users to register unique, human-readable identifiers and associate public-keys without any central or trusted party.

BNS consists of four layers, each of which can evolve independently. BlockChain Layer and Virtualchain Layer belong to the control plane, while Routing Layer and Storage Layer belong to the data plane. The architecture of BNS is shown in Fig. 19. (1) BlockChain Layer: Located at the bottom of the system, it is used to store Blockstack operations and provide consensus when writing operations. (2) Virtualchain Layer: The operations on identifiers and the state of them are defined as state machines in this layer. This layer can implement identifier registration, resolution, and management functions by defining new operations and states. Besides, this layer stores the identifier, the corresponding public key, and the hash value of the zone file. Among them, the public key is used to verify the signature of the data owner, and the hash value of zone file is used to retrieve the zone file in the peer network and verify the integrity of the zone file. (3) Routing Layer: Similar to DNS, BNS stores routing information through zone files. The difference is that in BNS, routing and actual storage of data are separated so that multiple storage methods can coexist. In the current implementation of BNS, hash value of zone file and the corresponding zone file is stored via DHT-based peer network [90]. And users do not need to trust the routing layer, because they can verify the integrity of zone files by hashing. (4) Storage Layer: This layer stores actual data. All stored data is signed by the data owner. And users do not need to trust the storage layer because they can verify signatures and data integrity at the control plane.

BNS Decouples the security of identifier registration and identifier ownership from the availability of data associated with identifiers by separating the control and data planes. And the Bootstrap trust problem is solved in a decentralized way by underlying blockchain to realize the joining of nodes without a trust center. Also, the identity and information integrity verification mechanism is provided. The designing of the Virtualchain layer enables defining new operations will not change the underlying blockchain. That can keep complexity and logic outside of blockchains. Moreover, BNS Separates routing from storage, so the choice on the actual storage system is unrestricted.

2) Identification Scheme: BNS is an alternative to DNS, which aims to remove the single root of DNS in a distributed service manner. Therefore, BNS does not provide a new identification format. In BNS, identifiers generate in a distributed way. And in order to avoid front-running where an attacker can race the user in registering the identifier, BNS designs a two-phase commit process. In this mechanism, the process of registering a identifier requires the user to go through two steps, preorder and register. The first user to successfully write

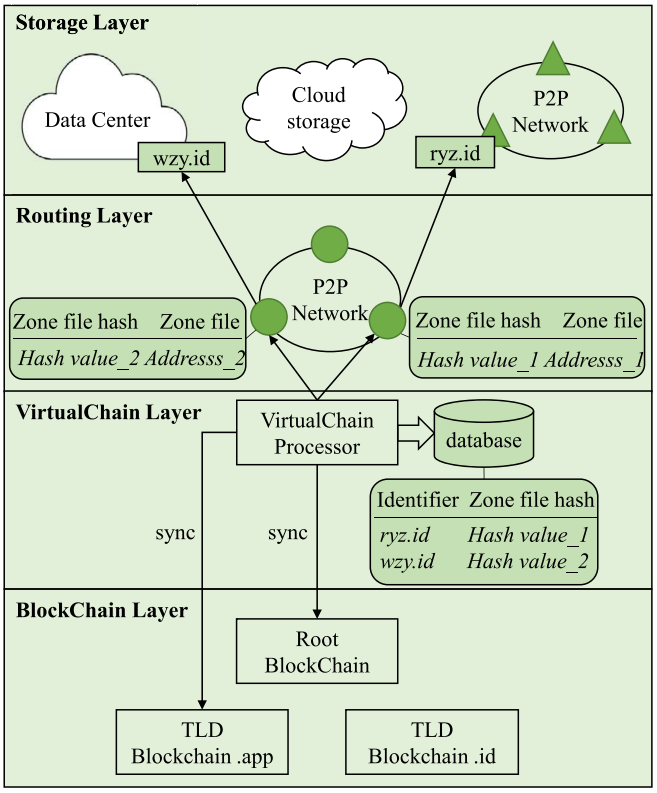


Fig. 19. The architecture of BNS.

both a preorder and a register transaction is granted ownership of the name.

3) Resolving: As a distributed alternative to DNS, BNS provides similar functionality with DNS but without any central root servers. The input to the BNS infrastructure is a human-readable identifier and then it returns an IP address or URL. And BNS stores routing information through zone files, and its zone file has the same format as DNS zone files. Specifically, BNS can be divided into three parts: BNS distributed infrastructure, trust zone, and peer network, as shown in Figure 20. (1) The BNS distributed infrastructure consists of multiple blockchains, and each blockchain corresponds to a identifier space. Similar to DNS, the information of top-level domains (identifier spaces) is registered on a root blockchain. Similarly, records in TLDs also point out the addresses of other blockchains registered in that TLD. (2) Trust zone refers to the local network, including end-users and local BNS servers. Among them, the local BNS server fetches data from the decentralized blockchain networks and keeps a local copy that is continuously synced with the blockchain networks. And it returns request responses to the end-user. (3) The Peer network stores the actual BNS zone files. It is worth mentioning that the Peer network is not within the trust zone. Users can verify the authenticity and integrity of zone files by the hash value of zone files.

4) Security: The security of BNS is mainly designed in two aspects, both of which are proposed for the architecture of distributed, non-central authority nodes. First of all, since there is no central identifier assignment and censorship institution,

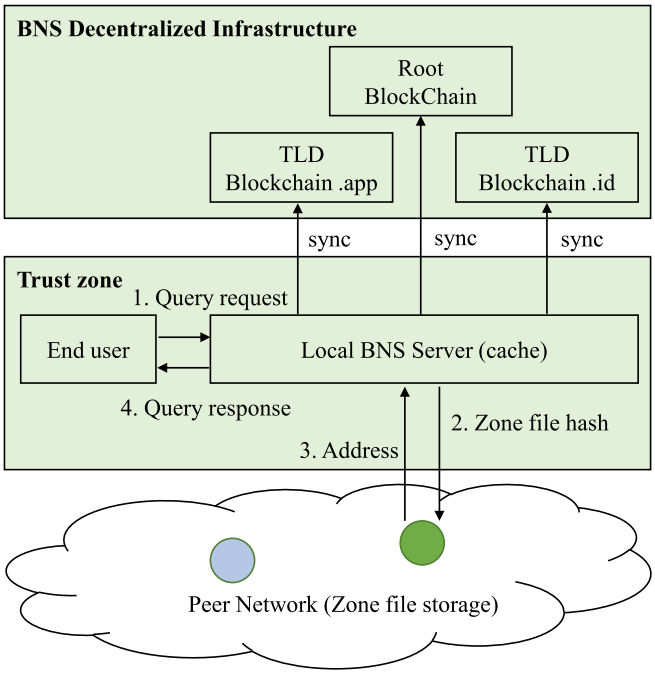


Fig. 20. The resolution process in BNS.

anyone can create a identifier space or register identifiers in a identifier space. So how to perform effective and secure identifier creation is a problem that needs to be solved. On the one hand, in order to avoid front-running where an attacker can race the user in registering the identifier, BNS designs a two-phase commit process. Only a user goes through two steps: preorder and register can register the identifier successfully. On the other hand, BNS specifies the expense of creating a identifier space or registering identifiers in by defining intelligent pricing functions. By defining pricing functions, BNS can not only guide the preference of naming of people but also prevent people from registering a lot of identifiers or identifier spaces that they do not intend to actually use. Secondly, since there is no trusted central node in the BNS, the trust zone only includes local networks that the user can control. So, in order to ensure that the information obtained by users is authentic, BNS provides a hash verification mechanism. Specifically, users can verify the authenticity and integrity of both zone files and data through hash operations.

5) Compatibility: First of all, as a complete replacement system of DNS, BNS has not yet provided a DNS-compatible solution. This means that if an end-user initiates a DNS query request, this user needs to send it to the Local DNS Server, but not to the Local BNS Server. In other words, the current Local BNS Server does not have the ability to recognize this is a DNS request, nor can it delegate the service to a DNS domain. In addition, because the format of the zone files of the BNS is the same as that of the DNS, the deployment of the BNS does not affect the DNS-based identity resolution systems such as OID and EPC.

G. Decentralized Identifiers

1) Overview: DID was proposed by the W3C DID Working Group in 2019. DID is a new, valuable, and worthy mention of a distributed identifier project that deserves continued follow-up. DID work is in full swing and has not yet been standardized. Currently, only drafts are published and comments are solicited, and the draft document may be updated, replaced or obsoleted by other documents at any time. So, we do not go into too much detail about this project but just talk about the highlights of the current version that is worth knowing.

DID is a new type of identifier to provide verifiable, decentralized digital identity. This new type of identifier can be registered, resolved, and used without any centralized registry, identity provider, or certificate authority. The DID system can be roughly divided into three parts: Client, DID Resolver, and Decentralized Identifier Registry, as shown in Fig. 21. First of all, the DID method is stored in the Decentralized Identifier Registry. In order to be able to declare a decentralized, globally unique method without a central node, DID gives a feasible solution, which is to use the blockchain's distributed ledger technology (DLT) for storage. Whenever a new DID method is generated, the consensus is required. Moreover, each DID Resolver can resolve multiple DID methods, and the client can invoke the DID Resolver through local/remote bindings. In addition, the DID resolver can also delegate some resolving functions to other DID resolvers, and multiple DID resolvers jointly provide the resolving service.

2) Identification Scheme: A DID is a globally unique identifier that does not require a centralized registration authority. And a DID is a simple text string consisting of three parts: URL scheme identifier (did), Identifier for the DID method, and DID method-specific identifier. Therefore, the key to ensuring that the DID is globally unique is to guarantee that the DID method identifier is globally unique. DID provides a reference implementation, that is, registering the DID method identifier in a DID method registry constructed by a distributed ledger or other decentralized networks. So that distributed users can form a consensus when a new DID method identifier generating. In addition, the system also provides a DID URL naming scheme. That is, more specific parameters such as service and version description can be added after the basic DID, so as to obtain more specific results in the resolving process.

3) Resolving: DID system can resolve the DID/DID URL into a DID document or more specific resolving results. Among them, if the input is DID, the process is called resolving in DID. While if the input is a DID URL, the process is called dereference. The specific resolving process of DID is shown in Fig. 21. (1) Client initiates a dereference request, assuming the DID URL is "*did : M1 :* 1234*; service = agent/ path? Query, options*". (2) After receiving the resolving request from the client, DID resolver R1 delegates part of the DID resolution to DID resolver R2 that supports resolving the DID method M1. (3) The DID resolver R2 sends a resolving request to the decentralized identifier registry containing M1 and then obtains the DID document corresponding to the DID. (4) DID resolver R2 returns the DID document to R1. (5) DID resolver R1 further dereferences the parameters of the DID URL and returns the site information of the requested service to client.

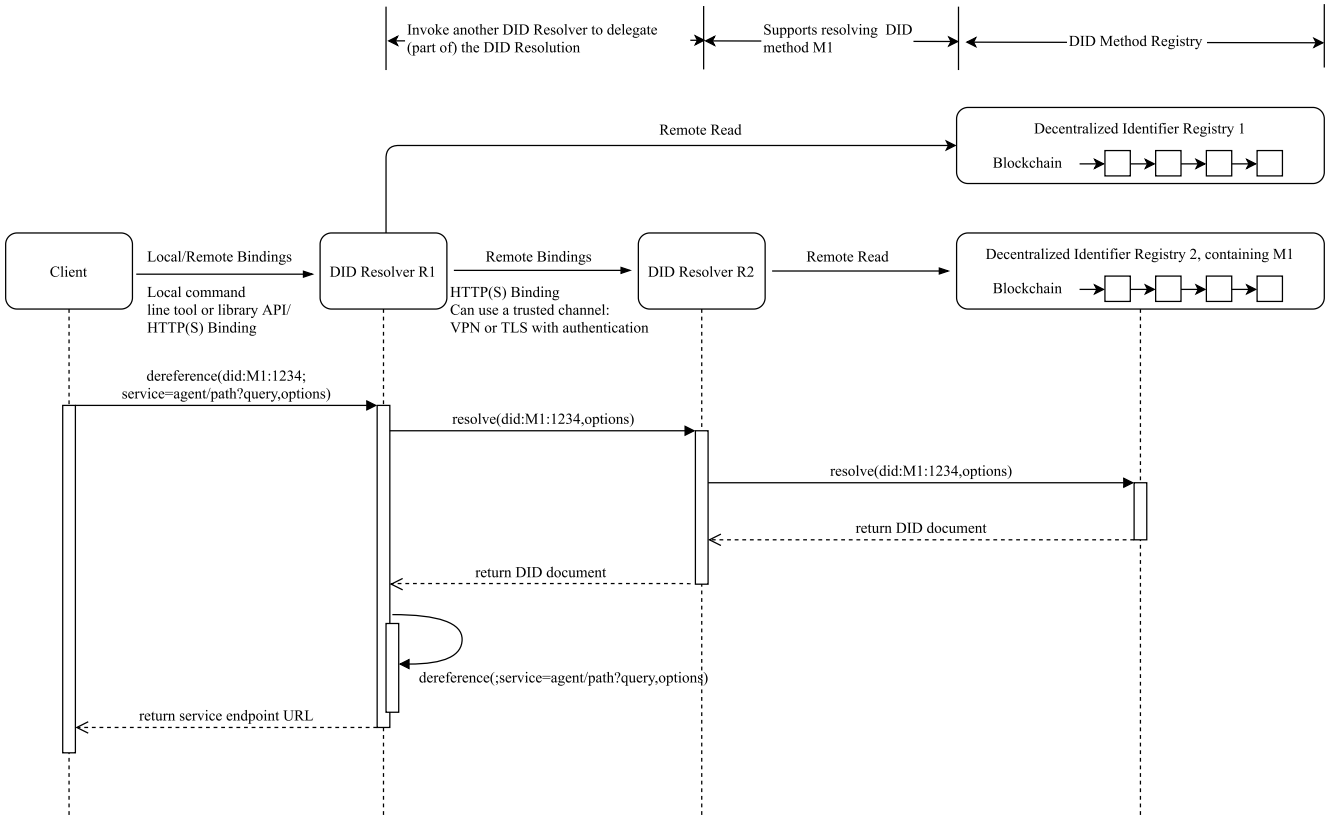


Fig. 21. The architecture and resolution process of DID.

4) Security: Since the distributed DID system does not have a central authority node, so how to ensure the security of it is very important. Firstly, the blockchain is difficult to tamper with and supports restoration after malicious updates. Moreover, the change of entry is undeniable. Secondly, users can verify whether DID documents have been tampered with through cryptography. In addition, users can authenticate the service endpoint. It is the responsibility of the service endpoint provider to comply with the requirements for authentication and to verify through the protocols supported by the service endpoint. Besides, DID supports deactivating DID files by replacing them with updated DID files. Also, DID supports a certain number of trusted parties to recover keys.

IV. Comparison of Systems

In this Section, we first compare each system based on the principle proposed in Section II-B. And we further discuss whether these systems could meet the needs of IIoT. We hope that the principle-based comparison can help readers choose the identity resolution system according to their requirements. Then, in Section IV-B, we compare systems based on the function-based framework proposed in Section II-C. We discuss the differences bring by different technology selections when each system realizes the same function. We hope this comparison can help readers better design their systems.

A. Principles-Based Comparison of Systems

In this part, we exam each system by the principles, as shown in Table VI and Table VII. And further, discuss whether they are suitable in IIoT and what the gaps are in meeting IIoT requirements. The work of DID is still in progress and may change at any time, so we do not discuss it too much.

1) Multi-Type Identity Subject Supporting: Which identity entities can be supported is mainly determined by the identity approach. The above systems all support multiple identity subjects. In addition, because GS1 and UID provide physical layer solutions, they are more used to identify physical objects.

2) Compatibility: In IIoT, enterprises usually already have their own internal identifying and resolving mechanism. All of the above systems can be compatible with existing enterprise identity resolution systems by assigning identity prefixes to enterprises. The difference is that if it is added to the OID, Handle, GNS, and BNS systems, the enterprise can keep its original identifier format in the subdomain, while if it is added to the EPC or UID system, the enterprise products need to re-identify according to the system rules.

3) Latency: First, from the perspective of architecture, EPC, OID, Handle, UID, and DID use a hierarchical resolution mechanism, so the cache can be fully utilized to reduce the resolution delay. while the resolving of GNS and BNS will use DHT, which may cause a large delay. In addition, the registration delay and resolution delay of BNS need to be considered separately. In BNS, reading the status of the identifier is fast and cheap, but writing the identifier is slow and

TABLE VI
PRINCIPLES-BASED COMPARISON OF SYSTEMS

System	Compatibility	Latency	Security and privacy
EPC	Poor compatibility, need to re-identify.	• Hierarchical resolution, the cache can reduce the resolution delay. • HTTP-based in second resolving stage, high delay.	• Tree structure, based on DNS, DDoS attacks may be an important issue. • DNS-based in first resolving stage, cannot support fine-grained access control. • DNS packets plain text transmission, privacy exposure. • Anti-physical replication, tamper, and forgery. • Central control has hidden dangers: information tampering, label copy, accountability difficult.
OID	Compatible with existing systems by assigning prefixes, original identifier format can be kept.	• Hierarchical resolution, the cache can reduce the resolution delay. • HTTP-based in second resolving stage, high delay.	• Tree structure, based on DNS, DDoS attacks may be an important issue. • DNS-based in first resolving stage, cannot support fine-grained access control. • DNS packets plain text transmission, privacy exposure.
Handle	Compatible with existing systems by assigning prefixes, original identifier format can be kept.	• Hierarchical resolution, the cache can reduce the resolution delay. • Resolving based on transport layer protocol, lower delay.	• Tree structure, DDoS attacks may be an important issue. • Authentication is required in core operations, while it is not required in resolving. • A secure session can be established, and the communication process within will be encrypted. • Central control has hidden dangers: information tampering, label copy, accountability difficult.
UID	Poor compatibility, need to re-identify.	• Hierarchical resolution, the cache can reduce the resolution delay. • Resolving based on transport layer protocol, lower delay.	• Tree structure, DDoS attacks may be an important issue • Provide Anti-physical replication, tamper, and forgery.
GNS	Compatible with existing systems by assigning prefixes, original identifier format can be kept.	• DHT routing, high delay. • Resolving based on transport layer protocol, lower delay.	• P2P networks in bottom layer, naturally resists DDoS. • GNS, people only with permissions and keys can get query results. • DHT-based resolving, the possibility of eavesdropping on the user's resolving request is reduced.
BNS	Compatible with existing systems by assigning prefixes, original identifier format can be kept.	• DHT routing, high delay. • Resolving identifiers is fast while registering is slow. • Two-phase commit process, registration is delay by several blocks. • Resolving based on transport layer protocol, lower delay.	• P2P networks in bottom layer, naturally resists DDoS. • DHT-based resolving, the possibility of eavesdropping on the user's resolving request is reduced.
DID	Compatible with existing systems by assigning prefixes, original identifier format can be kept.	• Hierarchical resolution mechanism, the cache can be fully utilized to reduce the resolution delay. • HTTP-based, high delay.	• Tree structure, DDoS attacks may be an important issue

expensive. Because registering and modifying the identifier requires one or more transactions to be sent to the underlying blockchain, and the BNS node will not process them until fully confirmed. In particular, in order to prevent name front-running, a two-phase commit process is used in BNS which needs two transactions. And only the pre-order transaction is confirmed by sequential blocks, the registration transaction can be confirmed. So the registration process should be delay by several blocks. Also, the throughput of the blockchain is limited. So the block generation interval is basically fixed and relatively long, resulting in a high delay for the identifier to be actually registered on the blockchain. The resolving delay is not much different from the normal reading of the database.

Secondly, from the perspective of protocol, the resolving in DID is HTTP-based. And requests in EPC and OID will be resolved in two steps, DNS resolution, and product information retrieval via HTTP based on URL. The HTTP is based on TCP connections, and the response delay is roughly the sum of the DNS domain name resolution time, the TCP connection establishment time, and the HTTP transaction time. The delay is relatively high. In contrast, handle, UID, GNS, and BNS are directly based on the transport layer, resulting in smaller message volume and lower delay. In particular, the Handle protocol is designed so that messages may be transmitted either as separate datagrams over UDP or as a continuous byte stream via a TCP connection. Therefore, the

TABLE VII
PRINCIPLES-BASED COMPARISON OF SYSTEMS (CONTINUED)

System	Fairness	Efficiency	Scalability	Customized service supporting
EPC	Tree structure, hierarchical data managment, service may be blocked.	• Centralized system, efficient in architecture. • HTTP in resolving, protocol lack efficiency. • Concise and efficient in resolving results (server address).	• Bounded identifier space, bottleneck in scalability. • Central tree-shaped architecture, single-point bottlenecks.	• Required HTTP, not friendly to resource-constrained terminals and sensors in IIoT. • The resolving result type cannot be customed.
OID	Tree structure, hierarchical data managment, service may be blocked.	• Centralized system, efficient in architecture. • HTTP in resolving, protocol lack efficiency. • Concise and efficient in resolving results (server address).	• Unbounded identifier space, scalable. • Central tree-shaped architecture, single-point bottlenecks.	• Required HTTP, not friendly to resource-constrained terminals and sensors in IIoT. • The resolving result type cannot be customed.
Handle	Tree structures, top layer consists of several parallel GHRs, fairer.	• Centralized system, efficient in architecture. • Transport layer protocols, more efficient. • Sacrifice effectiveness in resolving results (a group value set).	• Unbounded identifier space, scalable. • Central tree-shaped architecture, single-point bottlenecks.	• Transport layer protocols, need to run a client program, not friendly for some users in IIoT. • Communication protocols customization, friendly to IIoT with multiple type terminals. • Resolving result type customization.
UID	Tree structure, hierarchical data managment, service may be blocked.	• Centralized system, efficient in architecture. • Transport layer protocols, more efficient.	• Bounded identifier space, bottleneck in scalability. • Central tree-shaped architecture, single-point bottlenecks.	• Transport layer protocols, need to run a client program, not friendly for some users in IIoT. • The resolving result type cannot be customed.
GNS	Graph structure, node is registered in more than one upper node, fairer.	• Graph structure, sacrifice effectiveness. • Transport layer protocols, more efficient. • P2P-based, complex in network maintenance. • Concise and efficient in resolving results (server address).	• Unbounded identifier space, scalable. • P2P-based, more scalable.	• Transport layer protocols, need to run a client program, not friendly for some users in IIoT. • The resolving result type cannot be customed.
BNS	Tree structure, top-level domains is registered on a root blockchain, no longer a root node, service cannot be blocked.	• P2P-based, complex in network maintenance. • Blockchain-based, lack effectiveness, expensive in consensus, requires more storage. • Transport layer protocols, more efficient. • Concise and efficient in resolving results (server address).	• Unbounded identifier space, scalable. • Blockchain-based, bottleneck in scalability (appended only, storage exhaustion as nodes increases, arduous in entire network consensus).	• Transport layer protocols, need to run a client program, not friendly for some users in IIoT. • The consensus mechanism and the append-only accounting mechanism will consume a lot of computing and storage, not all IIoT nodes can support. • The resolving result type cannot be customed.
DID	Clients can query multiple DID resolvers and compare results, solve the service blocking to some certain.	• HTTP in resolving, protocol lack efficiency. • Concise and efficient in resolving results (server address). • Blockchain-based, lack effectiveness, expensive in consensus, requires more storage.	• Unbounded identifier space, scalable. • Blockchain-based, bottleneck in scalability (appended only, storage exhaustion as nodes increases, arduous in entire network consensus).	• Required HTTP, not friendly to resource-constrained terminals and sensors in IIoT. • The consensus mechanism and the append-only accounting mechanism will consume a lot of computing and storage, not all IIoT nodes can support.

service provider can use TCP in the management phase and UDP in the resolving phase with frequent queries, thereby reducing service delay.

4) Security and Privacy: First of all, In IIoT, there are widespread distributed edge devices and users, and DDoS attacks may be an important issue for EPC, OID, Handle, UID and DID that use a tree structure [94]. And as the upper application of DNS, EPC and OID may also face the risks brought by DNS. Because DNSSEC just could provide authentication and integrity, and does not protect DDoS attacks so these type of attacks are still very much a security issue when it comes to DNS [95]–[97]. And there is not any evidence to date that the GHR can handle a similar load in addition to protecting against the massive DDoS attacks seen on the Internet today [98]. In contrast, the bottom layers of P2P-based GNS and blockchain-based BNS are both P2P

networks, which naturally resists DDoS, but makes it easier to hide the traces of attackers [99]. Fortunately, decentralized architecture reduces the damage of DDoS. Especially for the blockchain, each node has complete information and can verify the validity of other nodes' data. Therefore, even if a certain node is attacked, the remaining nodes can maintain the entire blockchain system normally, which can effectively resist DDoS. Also, the blockchain uses a consensus mechanism to replace the central authentication mechanism. If the user authentication function of the traditional system is attacked, all user data may be modified. Blockchain does not require a third-party trust platform. Attackers must control at least 51% of the computing power of the entire network to tamper with data, which greatly increases the cost and difficulty of the attack. Moreover, in EPC and OID, the system is mainly controlled by government departments or a core enterprise. This kind of central control system has the following hidden dangers: information tampering, label copy, accountability difficult [100].

Secondly, DNS responds equally to everyone's request. Therefore, all data in DNS is visible and cannot support confidentiality, access control, or other services that can distinguish user identities. This may hinder the further application of EPC and OID in IIoT. Because in IIoT, fine-grained access control design allows enterprises to open data to its partner organizations more flexibly. In contrast, by default, the Handle client does not require any authentication when resolving. The client must be authenticated when requesting private data and performing remote management operations. The handle system uses the same tools for secure communications such as PKI, TLS, SSH, etc [101]. And in GNS, its resource records are encrypted, meaning that only people with permissions and keys can get query results.

Then, systems such as EPC and OID that are based on DNS face the risk of privacy exposure. Because DNS packets are transmitted on the network in plain text, the attacker can see which identifier the user requests to resolve. Therefore, the resolving operation may reveal business secrets such as the core production process. In the Handle system, the client can establish a secure session with the server, and the communication process within the same session will be encrypted. Moreover, the resolving process of GNS is based on DHT routing, and the transmission of information is scattered between nodes without passing through a centralized node. Therefore, the possibility of eavesdropping on the user's resolving request is reduced, thereby greatly improving the flexibility and reliability of the anonymous request.

Besides, EPC and UID also provide Anti-physical replication, tamper, and forgery.

5) Fairness: Fairness is to evaluate whether the services provided are neutral and non-discriminatory. The fairness of a system mainly depends on the system architecture and the resolving approach. In general, if the service of some nodes will fail due to being blocked by other nodes during the resolving process, then the fairness of this system needs to be improved. Service block in IIoT will bring more serious consequences than consumer IoT [53].

EPC, OID, and UID are all tree structures, and their data is managed in the multilevel hierarchical structure, the information server address is not registered directly in the data entry. The resolution server address which manages the data file of a lower layer to be searched next is registered in the data file of an upper layer. Therefore, the upper-layer nodes can block the services of the lower-layer nodes, which is unacceptable in IIoT. Also, Handle use a tree structure, but its top layer consists of several parallel GHRs. The GHRs are equal and communicate with each other to synchronize data. So, Handle is fairer than the above systems.

In contrast, although domain delegation is also required when resolving hierarchical identifiers, GNS uses a graph structure, which is fairer than the tree structure. In the tree structure, all the resolving requests of child nodes are forwarded from just one upper node. While in GNS, the sub-identifier-spaces node is registered in more than one upper node. When an upper node refuses to forward requests to a lower node, this function can be completed by other nodes (paths). This is equivalent to weakening the control of the upper node to the lower node, making the power gap between nodes smaller, avoiding possible single-point bottlenecks, and promoting the provision of more unbiased services.

Even if BNS is a tree structure, it is fairer than a traditional tree structure based system. Because in BNS the information for top-level domains (identifier spaces) is registered on a root blockchain, no longer a root node. So, the upper node cannot block the user's service request to the lower node.

Delegation to multiple DID resolvers in the sequence is needed when DID resolving. But how these DID resolvers managed does not mention yet in the current version. So, it is difficult to exam whether the service may be blocked. Besides, a client could query multiple DID resolvers and compare results, which can solve the unfairness caused by a single point to some certain.

6) Efficiency: In general, from an architectural perspective, EPC, OID, UID, Handle are more efficient as centralized systems. But EPC, OID, and DID all use HTTP in the resolving process. HTTP encapsulates the header on the transport layer protocol, which generates additional overhead.

From the perspective of resolving results, the output of EPC, OID, GNS, BNS, and DID are all server address, which is more concise and efficient. The resolving result of Handle supports customization and the metadata is well defined, while some effectiveness is sacrificed.

GNS based on graph structure is more robust and fair at the cost of effectiveness. The same lower-level node will be registered in multiple upper-level nodes, which is redundancy. Also, the bottom of P2P-based GNS and blockchain-based BNS and DID are P2P networks. The maintenance mechanism of the P2P network is complex, especially when the network fluctuates caused by frequent addition and withdrawal of nodes.

Besides, when the blockchain-based BNS is applied in IIoT, its lack of effectiveness is worth considering. The consensus mechanism of the blockchain is expensive. And compared with the traditional centralized database, it also requires more storage.

7) Scalability: In the future, a large number of devices will be accessed to IIoT. First of all, the size of the identifier space will affect the scalability. The identifier space size of EPC and UID is fixed, which may become a bottleneck for system expansion, while that of the other systems mentioned above is no limitation.

From an architectural perspective, in the IIoT scenario, scalability may be an important issue for a central tree-shaped system due to single-point bottlenecks such as EPC, OID, Handle, and UID. In particular in the fast-developing of edge computing and storage era.

The P2P-based GNS is more scalable. In the P2P network, each node is both a server and a client, which reduces the requirements for computing and storage capabilities of traditional C/S structure servers. In P2P networks, as users join, not only the demanding increases but the capabilities of the overall system are expanding simultaneously. At the same time, because resources and services are distributed across multiple nodes, the load balance of the entire system can be realized.

When BNS and DID based on blockchain are applied to IIoT, there may be a bottleneck in scalability. On the one hand, the blockchain can only be appended, and historical data is fully recorded. This mechanism provides decentralized security but also affects the scalability of the blockchain. As the number of nodes and transactions increases, blocks will grow rapidly, occupying a large amount of storage on nodes. On the other hand, as the core of the blockchain, the consensus mechanism determines the scalability at the basic level. When the scale of the blockchain network is large, the consensus of the entire network may become an arduous task. Reference [102] studied the scalability of the blockchain and analyzed the transaction throughput rate and transaction processing scalability of multiple consensus mechanisms.

8) Customized Service Supporting: From the perspective of the protocol, to obtain the information corresponding to the identifier in the EPC, OID, or DID system, the HTTP is required. However, the identity subjects and service providers in IIoT may be resource-constrained terminals and sensors, and may not all support HTTP transmission.

Handle, UID, GNS and BNS use transport layer protocols. To support the transport layer protocol, a client program needs to be run, which is not friendly for some users in IIoT. For example, users only have a personal computer, their pre-installed browsers can support HTTP. However, a new program needs to be installed to support Handle, GNS, and other systems based on transport layer protocols. Also, the consensus mechanism and the append-only accounting mechanism of the blockchain will consume a lot of computing and storage resources, and not all IIoT nodes can provide it.

In particular, the Handle system supports the customization of communication protocols, including TCP, UDP, and HTTP. The service provider can specify the protocol used by setting a value in the resolving result. For example, a resolving result may state the service site is composed of 3 servers. Among them, server 1 listening two ports 2641 and 2642. And it uses TCP/UDP for resolving and TCP for management. The customed transmission protocol is more friendly to IIoT scenarios with multiple type terminals. Moreover, different protocols correspond to different delays and costs, which can be chosen to meet different industry service requirements.

From the perspective of resolving results type, the Handle system supports the resolving result type customization to meet the specific service requirements. The mapping data of other systems can only be addresses such as URL or IP, and cannot meet the specific needs of services in terms of protocol, delay, overhead, and resolution content.

9) Lessons Learned: Key lessons learned from discussing whether systems can meet the IIoT principles are summarized below.

- The service delay will be affected by both the system architecture and the communication protocol. Centralized systems are generally faster than decentralized systems. Moreover, systems based on the transport layer protocol have lower latency than HTTP based. And it is better to use UDP for resolution with large volume and low-reliability requirements, while TCP/HTTP can be used for transmitting management information.
- Fairness depends on the structure of the system. The tree system lacks fairness because the upper nodes can block the lower nodes. The system of graph or net structure is fairer, which is mainly realized by DHT or blockchain.
- The structure, transmission protocol, and resolving results all affect the effectiveness of the system. Distributed systems generate a lot of cost for distributed consensus, sacrificing effectiveness. From a protocol perspective, because HTTP encapsulates headers and brings additional overhead, the efficiency is lower than that of transport layer protocols. The current types of resolving results can be divided into information server addresses and predefined metadata. It is more concise and efficient to use the addresses as resolving results.
- When measuring the scalability of a system, the size of the namespace and system structure should be considered. If the size of the namespace is fixed, it may become a bottleneck when IIoT develops rapidly in the future. Besides, a tree-structured system may have poor scalability due to a single point bottleneck. And blockchain-based systems may face scalability challenges due to limited throughput. At present, it seems that the system with better scalability is constructed using DHT.
- The system can support service customization by supporting transmission protocol and resolution type selection. The transmission protocol affects the performance of the service and the types of terminals that can use the service. Enterprises should choose the appropriate protocol and define the corresponding resolving result type according to the scenario.

B. Comparison of Systems Based on Technical Support for Key Functions

In this part, we present a systematic comparison of the mentioned systems from the perspective of technology selection, using DNS as a benchmark. A system always has natural pros and cons, which are often determined by its supporting

TABLE VIII
SUMMARY OF THE BASIC INFORMATION FOR THE COMPARED SYSTEMS, USING DNS AS A REFERENCE

System	The progress	Input and output	Key support technologies	Core features	Application
DNS	Standardization	Mapping domain name to IP	-	• Centralized • Already mature and easy to deploy	• Internet
EPC	Standardization	Mapping identifier to URL	-	• Centralized • Already mature and easy to deploy	• Automatic warehouse management • Product logistics tracking • Automatic supply chain management • Product assembly and production management • Product anti-counterfeiting
OID	Standardization	Mapping identifier to URL or IP	-	• Centralized • Already mature and easy to deploy	• Certificate Authority • Medical and health • Finance • Logistics • Food traceability
Handle	Standardization	Mapping identifier to customized type data	-	• Centralized • Already mature and easy to deploy • Design of parallel sites	• Digital Libraries Initiative (DLI) • Digital Object Unique Identifier (DOI)
UID	Standardization	Mapping identifier to context description	-	• Centralized • Already mature and easy to deploy • Design of ucR Graph	• TRON
GNS	Project, under standardization	Mapping identifier to public key or IP	DHT	• Decentralized • Having specific requirements for the underlying network, making deployment difficult in current networks	• GNUnet
BNS	Project	Mapping identifier to IP	DHT, blockchain	• Decentralized	• Internet
DID	Project, under standardization	Mapping identifier to URL	Blockchain	• Decentralized	-

technology. The characteristics are not absolute. The natural disadvantages may be insignificant in a certain application, and its advantages are exactly what the application needs.

This part can help researchers understand the functional composition and technical selection of an identity resolution system. And it can further inspire researchers to make tradeoffs based on requirements and design their systems flexible for a specific IIoT application. Since the work of DID is still in progress, this section also will not describe its details too much.

1) Basic Information: The basic information of the compared system is shown in Table VIII. These systems are at different stages of development. Among them, EPC, OID, Handle, and UID has been standardized and relatively mature. As new projects, GNS, BNS, and DID are still being developed, and GNS and DID are undergoing standardization work. Although each system has different objectives and application scenarios, they have similar key functions, as described in Section II. At the same time, even though the common function of the presented systems is generating, assigning and managing identity and resolving the corresponding content, each system achieves these functions from different ways. As described above, EPC, OID, Handle, and UID are centralized architectures, which are easier to implement. And as a new trend, GNS, BNS, and DID adopt a decentralized structure, which is a challenge to the existing technology and is not easy to achieve. GNS, BNS, and DID all adopt distributed technology to realize their respective functions, mainly including DHT and blockchain. Further, it is worth mentioning that, as a GNUnet naming service, GNS has requirements for the underlying network and needs to develop node software, so it is difficult to deploy in existing networks.

2) Identification Scheme: Identifying objects is a core function of the identity resolution system and has a profound impact on all aspects of the system. As mentioned above, there are two types of identifier format, that is, hierarchical or flat identifiers. And each approach has its own characteristics. Hierarchical identifiers are often human-readable, easy to aggregate, and convenient for domain delegation. Flat

TABLE IX
SUMMARY OF THE IDENTIFICATION SCHEME FOR THE COMPARED SYSTEMS, USING DNS AS A REFERENCE

System	Identifier format	The way identifiers generated	Identify Object	Identifier composition	Sub-identifier space	Identifier space size	Efficiency	Characteristic
DNS	Hierarchical	Centralized	Hosts	Characters	Unfixed length	Unbounded identifier space	Some redundant descriptions	Global, readable
EPC	Hierarchical	Centralized	Physical objects	Only numbers	Fixed length	Bounded identifier space	No redundant information, only the index is described without the information of the item itself	Global, hard to read
OID	Hierarchical	Centralized	Any physical or logical objects	Characters	Unfixed length	Unbounded identifier space	Multiple identify methods corresponding to different redundancy	Global, multiple identify methods corresponding to different readability
Handle	Prefix: hierarchial, suffix: either hierarchical or flat	Centralized	Digital object	Characters	Unfixed length	Unbounded identifier space	Unreadable prefix and customized suffix, less redundant description of the prefix. The efficiency of the identifier is uncertain.	The prefix is global and human unreadable, the suffix is local
UID	Hierarchical	Centralized	Physical or logical objects and the relationships between them	Only numbers	Fixed length	Bounded identifier space	No redundant information. Collect ucR Units from multiple distributed nodes, lack of efficiency	Global, hard to read
GNS	Hierarchical	Decentralized	Users, things and organizations	Public key: a 256-bit ECDSA key. Nickname: a UTF-8 string. Petname: customized	Public key: fixed length. Nickname: unfixed length. Petname: unfixed length	Public key: bounded identifier space. Nickname: unbounded identifier space. Petname: unbounded identifier space	Public key used for addressing: no redundant information. User-friendly names: description contains redundant information without affecting addressing performance	Public key: global, security, hard to read. Nickname: global, leak of security, readable. Petname: local, security, readable
BNS	Hierarchical	Decentralized	Hosts	Characters	Unfixed length	Unbounded identifier space	Some redundant descriptions	Global, readable
DID	Hierarchical	Decentralized	Digital object	-	-	-	-	-

identifiers can avoid location-identity binding, which is more secure. According to Table IX, it can be found that almost all the mentioned systems use hierarchical identifiers, which is easier to implement. And in academic research, flat identifiers can be seen a lot more [103]–[105]. In addition, in the Handle system, the identifier is composed of a prefix and a suffix, where the prefix is hierarchical and the suffix is customized, so it can be either hierarchical or flat.

A point worth discussing is how the identifier is generated. Generally speaking, the method of generating identifiers is closely related to the architecture of the system. Centralized systems often use centralized identifier generation methods, while decentralized systems often use decentralized identifier generation methods. Specifically, a centralized identifier generation method means that an organization that wants to obtain an identifier needs to submit an application to a public authoritative server. When the application is approved, the new identifier will be registered under the authoritative server as a subdomain of the identifier space. While in decentralized identifier generation methods, such as GNS, BNS, and DID, identifiers are not assigned by authoritative service nodes, but 1) generated by technical means that do not cause conflicts, such as public keys. 2) Obtained through consensus, such as blockchain. Obviously, the decentralized identifier generation method is more difficult to implement because it needs to consider a lot of issues, such as how to resolve conflicts, cybersquatting, and how to let other users know the existence of an identifier.

Another issue that needs to be discussed is the identifier space, which can be bounded or unbounded. If it

TABLE X
SUMMARY OF THE RESOLVING APPROACH FOR THE COMPARED SYSTEMS, USING DNS AS A REFERENCE

System	Whether DNS is required for resolution	Resolution approach	The structure of resolution system	Mapping data type	Efficiency
DNS	-	Recursion, iteration	Tree-shaped, multi-layer, single root	IP address	• Resolving the identifier of the unfixed length, the matching speed is slow • Support for resolving record aggregation
EPC	✔	Iteration	Tree-shaped, multi-layer, single root	URL, URI	• Resolving the identifier of the fixed length, the matching speed is fast • Support for resolving record aggregation
OID	✔	Recursion	Tree-shaped, multi-layer, single root	URL or IP address	• Resolving the identifier of the unfixed length, the matching speed is slow • Support for resolving record aggregation
Handle	✗	Iteration	Tree-shaped, two layers, each service node is consisting of multiple parallel service site	Customized mapping data	• Prefix: Matching fast • Suffix: Depends on the specific implementation • Support for resolving record aggregation
UID	✗	Recursion	Tree-shaped, two layers, distributed	Context description	• Resolving the identifier of the fixed length, the matching speed is fast • Support for resolving record aggregation
GNS	✗	Iteration	Directed graph	Public key or IP address	• Querying records through the DHT algorithm, slow resolving speed • Support for resolving record aggregation
BNS	✗	Iterative	Tree-shaped, Each node is implemented by a blockchain	IP address	• Querying records through the DHT algorithm, slow resolving speed • Support for resolving record aggregation
DID	✔	Recursion	Tree-shaped	URL	-

is a hierarchical identifier, bounded identifier space means that it consists of multiple fixed sub-identifier spaces. Correspondingly, unbounded identifier spaces often consist of multiple unfixed sub-identifier spaces. Identifier spaces for flat identifiers are generally bounded. A major argument about identifier spaces is the trade-off between capacity and efficiency. Fixed-length identifiers can be matched faster, but the number of identifiable objects is limited, this is the opposite of unfixed-length identifiers. In addition, each sub-identifier space can be composed of characters or numbers. The two choices mainly depend on the trade-off between human readability and efficiency. Character-based identifiers are often easier for humans to understand, but they bring information redundancy, that is, the description is not effective enough. While number-based identifiers are closer to the machine code and therefore more efficient.

3) Resolving Approach: The resolving approach of the compared system is shown in Table X. EPC, OID, and DID are upper-level applications of DNS systems on digital objects and resources. So they can take advantage of existing network infrastructure, and easy to deploy. However, because the resolution process of them depends on DNS, upgrades, replacements, or failures of the DNS system may cause their services to fail to provide. Secondly, this type of system inherits the native problems of DNS, such as the single point of failure, overload, and easy to be kidnapped by special organizations through legal or technical means. In addition, as an important infrastructure of the Internet, any extension of the DNS should be extra cautious. Because DNS is close related to the normal operation of the Internet. Finally, DNS is already overburdened, and DNS-based resolution services will cause a large number of requests to flood into DNS, which will further affect the operation of the DNS.

Another point to discuss is the structure of the resolving system. Almost all systems surveyed in this article use a tree-shaped structure, except that the GNS uses a graph-shaped structure. In the tree-shaped structure, a sub-identifier space is only registered in one parent identifier space, and the registration direction is one-way. While in the graph-shaped structure, a sub-identifier space can be registered in multiple parent identifier spaces, and supports two-way registration. The main difference between these two structures is robustness. Obviously, the graph-shaped structure is more robust, but it will also be more complicated.

In addition, the form of the resolving result, that is, the type of mapping data determines the application scenario of each system. Different types of mapping data cause the following differences in the identity resolution system: 1) Whether it

is friendly to manufacture companies. Most of the reviewed systems return an address in the form of IP or URL. This type of mapping data is more common but slightly rigid. There are many types of events and data streams in IIoT, such as latitude, longitude, temperature, humidity, etc. The output of the identity resolution system is a URL, which means that the metadata needs to be defined by the enterprise itself, which is equivalent to shifting the complexity from the network provider to the edge service provider. In particular, unlike the service providers on the Internet, most of the companies joining IIoT are manufacturing companies, who are not computer and Internet experts. They want to have a solution directly. Shielding the complexity of manufacturing companies can better popularize and develop IIoT. From this point of view, the most flexible of these systems is Handle, which supports the customization of resolving result types, so its application scenarios are more extensive. And it is convenient to shield the complexity of metadata definition for enterprises by formulating industry standards. Currently, Handle has been deployed as a key technology for IIoT in China. 2) Resolving efficiency. When resolving, systems such as EPC and OID need to map the identifier to a URL first, and then send a request to the URL to obtain data. By contrast, Handle only needs to send one request to get the corresponding value set. 3) Whether the identifier is persistent. Unlike IoT, in IIoT, the life cycle of the device is longer (over 15 years) [53]. So, long-term identification of some resources is crucial in IIoT. The handle system supports persistent references to objects [106]. On the contrary, the resolving result of EPC and OID is URL, and its value will change with the change of resource storage location, which may cause resolving failure. It is also worth mentioning that in IIoT, the spatiotemporal relationship between objects is more valuable than a single object, that is, an object should not be represented by a single node, but should be represented by a graph of nodes and edges [34]. The mapping data of UID is a context description, which can describe objects and their relationships, but other systems cannot. They can only describe the characteristics and attributes of a single object.

Finally, we discuss from the perspective of efficiency. The resolving approach of these systems corresponds to their identity approach. On the one hand, most of the above systems use a hierarchical format, so the records they use for resolving can be aggregated, which can reduce the storage burden. On the other hand, the resolution speed of fixed-length identifiers and non-fixed-length identifiers is different. Besides, GNS and BNS are distributed identity resolution systems, and their resolving processes are based on DHT, so the capacity of resolving services is larger, but the speed is slightly slower.

4) Security and Privacy: The security and privacy of the compared system are shown in Table XI. For an identity resolution system, end security, data security, operational security, and privacy are the keys that need to be guaranteed, as described in Section II. Almost all system surveyed provides functions such as authentication, data integrity verification, and encryption in different ways. In terms of security, corresponding to their respective architectures, the security mechanisms of these systems can also be divided into centralized and decentralized. In decentralized and distributed identity resolution systems, such as GNS, BNS, and DID, because there is no trusted server, the focus is on the confidentiality and integrity of data, which is fundamentally different in design concept from a centralized system. The main methods include providing self-authentication through a hash operation or using the blockchain to ensure that core data cannot be tampered with. In addition, because EPC and OID are upper-level applications of DNS systems on digital objects and resources. Therefore, EPC and OID inherit the security threats faced by DNS itself. Although the DNS security scheme DNSSEC was standardized in 1997, it is still far from widespread due to the distributed characteristics of the Internet [107].

In terms of privacy, the identity query request in the IIoT may reveal the information of the querier, including business partnerships, investment preferences, etc. However, EPC and OID are upper-level applications of DNS systems on digital objects and resources. DNS requests are sent in clear text, so the listener can know which domain name the user has queried. In order to solve this problem, DNS over TLS [108], [109]/HTTPS [110] (DoT/DoH) was proposed. They encrypt DNS requests to protect user privacy. The main difference between them is the different ports used. DoT has its own port 853, while DoH uses port 443, the standard HTTPS port. Although DoT/DoH can solve privacy issues to a certain extent, they are still in the proposal stage. And, because DoT uses a dedicated port, the use of DoT can be seen or even blocked. Besides, because DoH is based on HTTPS, which requires multiple data transfers to complete the protocol initialization. Therefore, the use of DoH will significantly increase the resolving process time-consuming. By contrast, the request and response in the Handle system can be encrypted. Although in EPC, OID, and other systems, the request sent to the URL is also encrypted. But on the one hand, they cannot guarantee the privacy of the DNS resolution stage, on the other hand, they implement privacy protection in different ways. The encryption function provided by the Handle system means that the complexity of the protocol is completed by Internet experts, while the encryption of the request sent to the URL transfers the complexity to the manufacturing company.

5) Compatibility and Deployment: Compatibility affects whether an identity resolution system can be really applied in actual scenarios. All of the above systems can compatible with existing enterprise information systems because they have designed local identifier spaces. Among them, The local identifier space of Handle can be completely customized, so the application scenarios are wider. While the identifiable object of UID and EPC will be limited by the format. Moreover, in order to actually use an identity resolution system, it is also necessary to consider how it is compatible with other existing identity resolution systems. To achieve multi-system compatibility, it may be helpful to build a suitable access platform, which is beyond the scope of this article. Finally, the adaptability of an identity resolution system to the future network architecture also needs to be considered. For example,

TABLE XI
SUMMARY OF THE SECURITY FOR THE COMPARED SYSTEMS, USING DNS AS A REFERENCE

System	Classification	Security and privacy
DNS	Centralized security mechanism	• DNSSEC. • DNS requests are sent in plain text.
EPC	Centralized security mechanism	• Application layer security mainly relies on DNS. • Provide security mechanisms for the exception layer. • DNS requests are sent in clear text. • The request sent to the url is encrypted
OID	Centralized security mechanism	• Can decide whether to use DNSSEC based on the security flags in the resolution request. • DNS requests are sent in clear text. • The request sent to the url is encrypted
Handle	Centralized security mechanism	• Does not rely on DNS. • Administrator and privileges design. • The security design of the Handle client. • The security design of the Handle server. • Resolving requests and responses can be encrypted.
UID	Centralized security mechanism	• To meet the differentiated security requirements of different applications, seven levels of security functions with fine-grained and flexible are designed.
GNS	Decentralized security mechanism	• Service nodes are isomorphic, so attacks on specific servers can be avoided. • The authenticity and integrity of data can be verified through the hash function. • Support key revocation. • Both queries and replies are encrypted.
BNS	Decentralized security mechanism	• Designs a two-phase commit proces to avoid identifier squatting. • The authenticity and integrity of data can be verified through the hash function. • The blockchain is difficult to tamper with.
DID	Decentralized security mechanism	• The authenticity and integrity of data can be verified through the hash function. • Supports for a certain number of trusted parties to recover keys. • The blockchain is difficult to tamper with.

some papers have explored the possibility of Handle being compatible with ICN.

6) Lessons Learned: Key lessons learned from discussing different implementations of system functions are summarized below.

- Generally, the namespace size of hierarchical identifiers is not fixed. In contrast, flat identifiers have a fixed namespace size. Flat identifiers are global and unreadable and are typically calculated by hash. The centralized identifier generation is relatively simple, but it is only suitable for centralized systems. Since there is no authoritative node for identifier allocation in a decentralized system, identifiers can only be generated in a distributed manner. There are two commonly used schemes to avoid the collision of distributed identifier generation, public keys based or blockchain-based. The identifier generated based on the public key is not human readable. The registration delay for blockchain-based identifier generation is generally longer.
- The structure of the system affects the robustness and delay of the resolution service. The central tree structure system may fail to resolve due to overload or node kidnapping. The distributed graph structure system is more robust, but the resolving delay is higher. Besides, DNS-based resolution can generally be divided into two steps. The first step is to convert the identifier into a domain name form and perform ordinary DNS resolution. The second step is to retrieve the information corresponding to the identifier further. Therefore, if the DNS-based resolution system is used in the future of the rapid development of IIoT, it may introduce a large amount of DNS traffic and affect the normal use of the Internet.
- Decentralized systems are more resistant to DDoS attacks than centralized systems. The security of blockchain-based systems is high. Besides, DNS-based systems face the risk of privacy exposure. On the one hand, DNS responds equally to everyone's request. Therefore, all data in DNS is visible and cannot support confidentiality. On the other hand, Since DNS packets are transmitted in plain text, the attacker can see which identifier the user requests to resolve.

V. Challenges and Future Research Directions

Despite some developments in the identity resolution system in IIoT, many significant research challenges remain to be addressed due to the requirement of decentralization, compatibility and performance. In this Section, we discuss some challenges and present some future research directions.

A. Decentralized Distributed Identity Resolution System

A promising direction in the future is to design decentralized distributed identity resolution systems. In IIoT, a large number of identifier resolution services and data sharing occur among peer-to-peer multi-stakeholders. Each of them may choose different identity resolution schemes internally, so the multi-party negotiation of transmission protocol and data format is needed. A decentralized Distributed identity resolution system generally provides a distributed consensus solution, which can naturally solve the above problems. Moreover, there are a large number of distributed devices at the edge of the IIoT, so it is logical to adopt distributed identity resolution technology.

Besides, the distributed identity resolution system can also overcome many problems of the traditional tree systems, such as the performance bottleneck and the single point of failure [39]. And it also has many advantages on scalability, availability, and reliability, and always autonomously controllable. It is more suitable for IIoT scenarios with massive access and higher requirements for security, fault tolerance, and reliability. Therefore, more and more researchers have begun to explore decentralized identity resolution systems. Three of the more influential projects, GNS, BNS, and DID, have been surveyed in this article.

The core issue of the decentralized distributed identity resolution system is how to achieve the following functions without a central node: 1) Identifiers can be generated in a distributed environment without causing a lot of waste of resources. That is, there is no conflict between identifiers without central identity distribution mechanisms. Also, mechanisms need to be designed to ensure that people do not register a bunch of identifiers they do not need. It is worth mentioning that algorithms capable of generating globally unique identifiers generate random strings that humans cannot understand. This can be read in the theory of identifiers in the Zooko triangle, that is, only two can be satisfied among human-readable, decentralized, and secure. 2) Resolving in a distributed environment. Specifically, it includes the construction of the relationship between service nodes, the coping methods of the dynamic online and offline nodes, and the provision of distributed caches to improve system performance, and the design of incentive mechanisms to achieve distributed caches. 3) Security in a distributed environment. That is, how to realize the true and complete transmission of information without a third-party security authority review agency (no trust zone).

In addition to the three projects mentioned in this article, there are also some researches on distributed identity resolution systems, most of which are based on DHT or blockchain. Reference [111] proposed a domain name system based on DHT technology. This system inherits the fault tolerance and load balance of DHT technology and can solve many management problems facing DNS. In [112], a DHT-based service architecture was proposed, which can replace the object naming service in response to the lack of robustness, complex configuration, and weak security of DNS. This solution can enhance the privacy protection of users to a certain extent. In [105], an identity resolution system based on a multi-level DHT algorithm was presented. This system stores identifiers and their mapping data in a multi-level DHT network, and DHT nodes provide resolving services. Reference [113] proposed a DHT-DNS hybrid domain name system. This system mounts the DHT namespace under the DNS tree, and the DHT node acts as an authoritative name server. When a resolution request arrives, it will be resolved to a DHT node. Reference [104] proposed an identity resolution scheme based on DHT-DNS. This solution uses the hash string to identify objects to provide heterogeneous identifiers compatibility. At the same time, a 0-1 binary tree is used to build a resolving architecture, in which each hash string can be mapped to a leaf node of the binary tree.

Although there are some exciting new results, the relevant research is still in its infancy, and existing a lot of problems have not been solved. Overall, decentralized systems are more complex to implement, and each user needs to install additional software. So how to integrate it into the IIoT with a large number of resource-constrained sensors is worth studying. Moreover, if the system is blockchain-based, it will face the limitation throughput naturally brought by the blockchain. In particular, name squatting and front-running are easier to occur due to no centralized control and high transaction latency [114]. While if it is based on P2P, it will also face the multi-party consensus difficulty and security issues caused by P2P openness.

B. Compatibility

There may be multiple heterogeneous resources that need to be identified within an industrial enterprise, including RFID, network devices, URNs. Therefore, when the identity resolution system is deployed, multiple heterogeneous resource compatibility solutions need to be considered. The current mainstream solution is to build several local resolving servers to manage multiple heterogeneous domains. The local resolving server mainly provides the following functions: 1) Provide a layer of overlay identifiers for multiple heterogeneous domains, which are then registered in the company's identity resolution system. And provide local identifier conversion when resolving. 2) Provide corresponding communication protocol according to resource type.

An identification compatibility scheme based on OID architecture was proposed in [115]. This scheme uses local identifiers and OID prefixes to form a Virtual Identifier Layer (VIL), by which the heterogeneous identifiers interoperability can be achieved. When the resolving request arrives, it is first routed to ORS, and then redirected to the corresponding resolving server to respond to the request. This solution can effectively achieve the compatibility and interoperability of heterogeneous identifiers, including barcode, RFID, network resources,

USNs. In [116], an OID-based heterogeneous identifier integrated resolving architecture was proposed. In this scheme, the local identifiers are managed by the ID registry, which is registered and applied in the ORS. After the resolving request reaches the ORS, it will be redirected to the local ID registry according to the mapping data in the ORS. Reference [114] proposes a UID-CoAP joint architecture based on UID technology and Constrained Application Protocol (CoAP), which can implement hosting services on general embedded nodes. Among them, CoAP is used for communication between resource-constrained nodes, and UID technology is responsible for describing the knowledge and data needed to implement IoT services. A UID-based ubiquitous computing interoperability architecture was proposed in [117]. In this architecture, an information server intermediary is established for each heterogeneous subspace, and the information server intermediary is registered in the ucode resolving server. When the resolving request arrives, the resolving server will map the ucode to the address of the information server intermediary, and then redirect the request based on the information server intermediary.

This kind of scheme is simple, but the architecture is complicated and the overhead is high. More important, industrial enterprises are required to design the architecture and transmission logic, which they are not good at. The Asset Administration Shell (AAS) [118] may be a solution in the future. Its purpose is to exchange asset-related data among industrial assets and between assets and production orchestration systems or engineering tools. But the research is still under exploration.

C. Deep Integration of Identity Resolution System and Specific Application

Deep integration is designing schemes for a specific application, including system selection, transmission protocol selection, and data format design. In IIoT, different applications have different service models and requirements on compatibility, performance, fairness and security. For example, when the identity resolution system is used to connect the information of multiple subsidiaries within a group, EPC and OID with an efficient hierarchical structure may be a good choice. Because their structure is simple and easy to deploy. Most importantly, as the parent node, the parent company has no incentive to block the services of the subsidiary. However, in cross-border trade, information needs to be shared among companies from different countries. In this scenario, service block risks and fairness issues need to be considered. Because most companies may not be willing to let an organization in another country assign identifiers and forward resolution requests for themselves. In this case, the Handle system with multiple parallel roots can be used. Or autonomous controllable identifiers systems GNS, BNS, and DID are also good choices. On the other hand, in delay-sensitive applications, it is best to resolve based on UDP. So, DID that requires HTTP may not be suitable. While in accuracy-sensitive applications, the TCP or HTTP can be chosen. Also, the number of subjects to be identified and their growth trends should match the size of the system identifier space. Besides, many distributed devices, storage, and computing resources at the edge of the network in IIoT need to be identified. It is more suitable to use distributed identifier generation approaches, such as GNS and BNS.

The standard of systems only describes their basic functions. When researchers choose one system to apply in an application, a specific solution needs to be given. For example, reference [119] proposed an anti-counterfeiting smart system based on Handle. The product identifiers are redesigned based on the Handle standard to reflect the logistics information. The logistic sequence is defined by a number of prefixes, from country prefix to seller's prefix. In short, in this application, the identifier format and Handle's each layer service's deployment are specified. If a fire occurs in the factory, the notification of fire agencies and police, and the opening of the escape system needs to be done simultaneously. This kind of complicated service cannot be provided by traditional identity resolution systems directly. Reference [120] proposed an OID-based resolution framework for service groups to solve this problem. A service group is a set consist of multiple specific services and is registered to ORS. An event, such as a fire, is also identified by OID. A service group can be called concurrently when an event is triggered to guarantee lower latency. The specific data transmission format and service invocation mechanism of this scheme are refined in specific applications, which beyond the standard scope of the identity resolution system itself.

Moreover, deep integration not only means system choosing but also covers function expansion and technology replacement. In this case, only part of the system's functions can be used, such as coding schemes. The function-based comparison in Section IV-B helps researchers know the advantages of each system and choose suitable technology in specific IIoT scenarios. For example, when the identity resolution system is deeply integrated with the food traceability, the main issue is anti-tampering and the decentralized multi-party information secure sharing. However, the traditional centralized traceability system has the following hidden danger: information tampering, accountability difficulty, and spamming products. In [100], a food traceability system based on blockchain and EPC is proposed. In this scheme, only the EPC coding is used, while the traditional ONS functions are replaced by blockchain and IPFS. In the traditional EPC system, ONS provides identifier registration and resolution. In [100], the identifiers are issued through the blockchain. And IPFS supports the main part of resolving service to ensure effectiveness and scalability. Another example, in a scenario where information is transmitted between multiple organizations, a hierarchical identity resolution system is inefficient and cumbersome. Reference [114] proposed a blockchain-based UID management scheme to solve the problems caused by the hierarchical structure. In the scheme, the registration and resolution mechanism of the traditional UID is replaced with the blockchain. Besides, to suit blockchain-based ucode allocation and be compatible with hierarchical UID, some adjustments of the ucode format in the scheme have also been made based on the standard.

In general, the cross-innovation and in-depth integration of the identity resolution system and the specific applications has a broad research scene.

D. Multi-Identity Resolution System Interoperability Scheme

At present, there are multiple identity resolution systems, and it is unrealistic for each company to register in each system. To maximize the value of data, the interoperability between multiple identity resolution systems is worth studying. In an ideal situation, users can resolve the identifiers in each system through a unified client. If the optimization and in-depth integration of the system and specific applications can be regarded as vertical optimization, the interoperability of the multi-identity resolution system can be regarded as horizontal expansion.

To interoperate between multiple identity resolution systems, the following two main functions need to be implemented, protocol conversion and data filling. A simple way is to deploy a proxy server between the user and the system. The proxy server serves as the protocol coordinator to complete the data exchange between the user and the system. The user first sends the request to the proxy server, the request will be re-encapsulates and fills in data. Then the request will be forward to the corresponding identity resolution system via the protocol supported by the system. After responding, the proxy server returns the response message through the protocol supported by the user. However, there are many types of users and service providers in IIoT, and they each support different transmission protocols. Therefore, the transmission protocol and data format between the user and the proxy server need to be negotiated. How to implement the above process efficiently in the IIoT where resource-constrained devices are widespread is a challenge. Also, the implementation of the proxy server will introduce a third-party "middleman" in communications, which may bring new security risks and privacy issues. Therefore, this field needs more extensive research in the future.

VI. Broader Perspectives

The development of the identity resolution system in the IIoT may be affected by many other technologies. In this section, we will briefly discuss these technologies and give a broader perspective, if the technology has one of the following characteristics: 1) The problems of it to be solved overlap with the construction of an identity resolution system in IIoT and its design method is worthy as a reference. 2)It can enable the identity resolution system. 3) It can optimize the identity resolution system. 4) It is an inevitable technological trend in the future, so it is necessary to consider its impact on the identity resolution system.

A. Self-Sovereign Identity

The increasing usage of different online services requires an efficient digital identity management approach. The internet lacks a layer of identity protocol and this shifts the responsibility for identification and verification to service providers (SP) [41]. The identity issue and authentication are provided by SP so that the SP and the identity provider (IdP) is combined [121]. This is highly inefficient due to the duplication of the information among multi-SP. And users do not have any control over their identity data. Besides, password-based authentication, data fragmentation, client on-boarding, and identity breaches also call for a new identity management model [122].

To solve the above issue, Self-Sovereign Identity (SSI) was introduced. It enabling identities, authentication, authorization, roles, and privileges individual managing, within an organization or across boundaries [122]. By SSI, users can fully own and control their identity data. Reference [121] has pointed out the stakeholder of an SSI system, including citizens, public administration, and businesses. It can be easily utilized for cross-border authentication and cross border services. And ten principles of SSI have been provided in [123].

SSI is regarded to have an influential effect on how we interact with each other over the future Internet. And the blockchain technology provides a good basis to create an SSI system. Reference [124] illustrates several envisioned flows to build SSI leveraging blockchain. And several specifications to evaluate any SSI solution was present in [125]. The three major SSI objects was identified in [126], and the fine-grained design patterns for blockchain based SSI was presented. An SSI solution for permissionless decentralized digitized passport was proposed in [127]. Reference [128] surveyed several blockchain-based SSI in healthcare.

Due to the following reasons, SSI can inspire the design of the identity resolution system in IIoT. 1) The scope of the problems to be solved overlaps. As a decentralized distributed identity management model, SSI needs to solve the distributed management, authentication, and verification of identifiers. This is consistent with the problems faced by the distributed identity resolution system in IIoT. Moreover, another important purpose of SSI is to solve the identity fragmentation, that is, how to integrate identity data in multiple systems to achieve cross-border authentication and cross border services. This is highly similar to the data interoperability of multiple identity resolution systems among multiple businesses in IIoT. Therefore, the distributed data sharing mechanism in SSI can be used for reference to promote the sharing and interoperability of heterogeneous data between inter-organizations. 2) The technology selection overlaps. In SSI, peer-to-peer technologies, such as blockchain, are used extensively to achieve self-control of identifiers. While building an identity resolution system in IIoT, or integrating systems into specific applications, blockchain has become an indispensable supporting technology. Therefore, the application of blockchain in SSI can bring inspiration. 3) The challenges faced are similar. Because the problems to be solved and technology selection of SSI overlap with the construction of an identity resolution system in IIoT, they also face similar problems. For example, the storage consumption and cost issues that blockchain and consensus mechanism brings.

Besides, there are some differences between SSI and the distributed identity resolution system in IIoT. 1) Different service models. SSI is user-centric, while the identity resolution system in IIoT is enterprise-centric. This means that the main

object of SSI registration is ordinary users, while in IIoT, more are enterprises. The computing resources, storage resources, and stability of ordinary users are not as good as those of enterprises. Therefore, the choice of the blockchain type and consensus mechanism will be different. 2) SSI is more innovative. in the current user identity ecosystem, the identity holders always not be the identity owners. So, in addition to the distributed data sharing mechanism, SSI also needs to consider how to let users control their identifiers. While in IIoT, products and devices data is held by enterprises. That is, the owner of the identifier and the holder of the identifier are unified. So, the identity resolution system in IIoT focuses more on how to promote the sharing and interoperability of heterogeneous data cross-organization. 3) Different use cases. SSI is mostly used for digital wallets, digital passports, and patient digital identity among multiple medical institutions currently. While the identity resolution system in IIoT is mainly used for product traceability and supply chain management. Therefore, when it is deeply integrated with specific applications, different specific problems need to be solved.

B. Information-Centric Networking

ICN is a promising networking paradigm, where data exchange is based on the name of the content, not the IP address of the endpoint [129]–[131]. Well-known ICN projects include CCN/NDN [132], PURSUIT [133], MobilityFirst [134], etc. As an important project in ICN, MobilityFirst supports hybrid name/address-based routing. So, compared with other ICN projects that routing based on names entirely, MobilityFirst is more similar to identity resolution systems. So in this part, we will take MobilityFirst as an example to analyze the enlightening effect of ICN on deployment identity resolution systems in IIoT.

ICN can inspire the design of identity resolution systems for the following reasons. 1) The input and output are similar. The input of ICN and IRS are both names, and the output is the mapping data attached to the name. For example, in the name resolution of MobilityFirst, the input is the name of a file or device, and the output is the corresponding network address. 2) Their core functions and supporting technologies overlap, so ICN can be used as a reference. The functions of identity resolution systems and ICN both involve naming, name resolution, and security [135]. For example, in MobilityFirst, names are basically public keys assigned by a name certification service, which is similar to the public key-based identification in GNS. And, in MobilityFirst, the dynamic binding between the name of a network object and its current network addresses is provided based on DHT [136], [137]. This is similar to the DHT-based resolving service in GNS. Besides, in MobilityFirst, deriving names as a cryptographic hash of a public key also enables them to be self-certifying. Likewise, in BNS, the zone file hash is used for addressing and integrity verification.

When designing an identity resolution system with reference to ICN, we also need to pay attention to their differences. ICN works at the network layer, while identity resolution systems in IIoT works at the application layer. This means ICN mainly focuses on routing protocol design and compatible with existing IP networks. While identity resolution systems mainly design business models and service architecture. Including enterprise network organization, micro-service architecture construction to better realize the identity resolution function. Moreover, identity resolution systems also need to design interfaces for enterprises and upper-layer applications, and select the appropriate communication protocol for the business.

C. Blockchain

DLT has attracted a lot of attention from academia and industry in recent years. DLT is a technology in which multiple distributed nodes participate in storing data, and each node can monitor the legality of the transaction and can also prove it. Blockchain is one form of it, providing trustworthy, transparent, tamper-resistant and consistent services by a group of nodes without a central authority. Each node in the blockchain uses a chain-based block structure to store complete data. Furthermore, each node in the blockchain is independent and of equal status. These nodes rely on a consensus mechanism to ensure storage consistency. In the blockchain, no node can write to the ledger independently, so avoiding false accounts caused by a single bookkeeper being controlled or bribed, thereby ensuring the security of account data. Nowadays, blockchain has been applied in many fields such as medical treatment [138], economics [139] and network [140]. It is usually used to support incentive or security mechanism for the distributed system [141], [142].

There are two main types of influence of the blockchain on the identity resolution system. One is to optimize the existing system by virtue of the non-centralized, tamper-resistant, and security features of the blockchain. And the information sharing among multiple organizations can also be promoted by the consensus mechanism. The other is to directly use the blockchain as an enabling technology to build a new identity resolution system.

From an optimization perspective, the introduction of blockchain can boost the process of identifier sharing and management of multiple related organizations. Also, the security of the system can be enhanced, including the status of the device that cannot be denied, and the history of each access operation is transparent to related organizations. Vulnerability to the single point of failure and data tampering is a critical issue for a centralized identity resolution system. Blockchain-based decentralization DNS data storage method has been proposed to solve this problem [143], [144], where the key information of the zone file resolution data is stored in multiple parallel DNS nodes. In [145], a new distributed domain name service ConsortiumDNS based on the consortium chain was proposed. In [146], a novel DNS cache resources trusted sharing model has been proposed, which can improve the credibility of DNS resolution results with the help of the consortium blockchain. In [147], [148], a blockchain-based ONS with a tokenized authority has been proposed, where the blockchain is used to strengthen the security of ONS. In [114], a ucode ownership management system based on blockchain has been proposed.

And further, a user-friendly and efficient ucode allocation method has been provided. In [149], a novel blockchain-based mutual authentication security protocol was proposed. It can improve the authentication between tags and readers to prevent potential attacks in RFID without trusted third parties. In [100], a food traceability system based on blockchain and EPC is proposed. The traditional ONS functions are replaced by blockchain and IPFS to overcome the tampering problem and accountability difficulty.

From the perspective of building new identity resolution systems, the blockchain can be used as an architecture supporting technology, such as the BNS surveyed in this article. Blockchain is an indispensable core technology for such identity resolution systems and is often used for distributed identifier generation, registration, security protection and providing incentives mechanism. Building new identity resolution systems based on blockchain has different scope between only use it to optimize existing systems. When optimizing an existing system, it is often focused on replacing the existing management mechanism with a blockchain, and it is also necessary to consider how to be compatible with the original system. while when building a new system, it is more necessary to provide solutions to the decentralized generation of identifiers, the squatting of identifiers, and the waste of identifier space.

D. Machine Learning

With the further use of the identity resolution system in IIoT, a large amount of data will be generated. Many applications in IIoT will generate resolving requests, such as food traceability, product life cycle management, and supply chain management. The analysis of these data can promote the development of IIoT services. Machine learning (ML) [150] is a good way to achieve the above goals, which is to improve the performance of specific systems by continuously learning data. It can make IIoT services smarter by processing the generated data.

In particular, as a public infrastructure in the IIoT, the identity resolution system must have the ability to perceive risks and defend against attacks. Based on big data analysis, security threats can be discovered, identified, understood, analyzed, and dealt with. Since the research on the identity resolution system in IIoT is still in the exploratory stage, as the most widely used identity resolution system on the Internet, DNS's past research ideas are worth learning from. A survey of systems that utilized passive DNS traffic to detect malicious behaviors on the Internet was presented in [48]. It highlighted the main strengths and weaknesses of the implemented systems through an in-depth analysis of the detection approach, collected data, and detection outcomes. Reference [151] proposed an advanced detection method against DNS cache poisoning attacks using machine learning techniques, where in addition to using the basic 5-tuple information of a DNS packet, a lot of special features were added to identify the DNS response packets used for cache poisoning attacks. In [152], a cognitive feature extraction model based on scaling and multifractal dimension trajectory to analyze Internet traffic time series was presented. In this way, DNS DoS attacks can be detected. Furthermore, some researches detect malicious activity and botnets by monitoring DNS traffic and logs, or building graph relationships between labeled domains which can be traced back from queries history of all domains [153]–[155]. Finally, in [156], an ML-based approach is presented to tackle the typosquatting vulnerability, where a majority voting-based ensemble learning classifier built using five classification algorithms is proposed that can detect with high accuracy. Besides, Reference [157] focuses on more basic issues, where a deep learning method, called Bytelevel CNN, to detect the DNS tunnels was proposed. It can solve the problem of manual feature extraction limitation and improve the detection accuracy of DNS tunnels.

In summary, Using ML in identity resolution systems can enhance the development of IoT services.

E. Software Defined Networking

Software-defined networking (SDN) [158] is a new network architecture proposed by Stanford University. SDN realizes flexible control of network traffic by separating the control plane of the network device from the data plane. SDN can make the network more intelligent and provide a good platform for innovation in core networks and applications. At present, the discussion and research on the application of SDN in IIoT is increasing. On the one hand, SDN can be applied in smart factories to achieve fast, automatic, and customized industrial network configuration. On the other hand, applying Software-Defined Wide-area Network (SDWAN) technology between multiple factories can enable dynamic routing of links, flexible deployment, and enhanced security.

At present, SDN has been applied in IIoT to solve various problems, including IIoT data center construction [159], adaptive transmission optimization [160], security [161], [162], and network resilience enhancement [163]. Therefore, SDN technology is likely to be a key technology for IIoT in the future. The overall scheduling of the application layer and the network layer has always been a research direction. Specifically, an integrated consideration of the identity resolution system and the underlying network architecture is conducive to improving performance. At present, there have been researches focusing on the comprehensive consideration of SDN and DNS. In [164], a comprehensive survey of defense mechanisms against DDoS attacks using SDN has been provided. Moreover, they review the researches about launching DDoS attacks on SDN, as well as the methods against DDoS attacks in SDN. In [165], a defense mechanism has been proposed, in which all SDN switches linking DNS servers monitor the speed of DNS request packets. It can easily detect the attacks, protect the victim quickly, then pinpoint all zombies and finally isolate them from the SDN network. In [166], a novel detection strategy protecting the DNS server by manipulating the Openflow control message has been presented. Whenever a server controller receives a query packet, it will send an authentication packet back to the client network to verify the legitimacy of the query. In addition, there are also studies exploring the combination of EP and SDN [167], [168]. The quality of service provided by SDN-based EPC networks

was evaluated in [167], which is essential for the successful delivery of real-time multimedia applications in mobile operator networks. In the three QoS indicators of delay, jitter, and packet loss, SDN-based EPC is significantly better than its traditional EPC. Similarly and more broadly, other identity resolution service in IIoT can also cooperate with SDN technology to reduce latency, increase service flexibility, security and robustness. For example, SDN switches can be used to monitor request packets to ensure security. Or SDN technology can be used to observe the state of the network to provide suggestions for caching of entries.

VII. Conclusion

The recent rapid development of network and industrial technology prosper industrial intelligent production. IIoT connects sensors, industrial equipment, products, and staff in the factory, enabling industrial processes monitor, automatic control, and costs optimization. In this article, we have provided a survey of potential identity resolution systems for IIoT, which is a core infrastructure in IIoT. We have begun our discussion with an overview of the identity resolution system and have presented why they are important for IIoT. Then, we have discussed the design principals of them. Then, we have provided key functions and properties for identity resolution systems, and give a universal reference framework to judge them. Besides, we have further presented the classification of existing identity resolution systems to give readers a rough understanding of the field. Next, we have discussed several existing influential systems and further compared them in terms of the design principles and technology selection. We also discussed the challenges and some important research directions of this field. Finally, we explored some broader perspectives.

The identity resolution system is important for IIoT since it facilitates the intercommunication between multiple isolated plants. And a manufacturing process can be described with the help of identity resolution systems. However, this field is still in its infancy and needs to be further explored. Especially the deep integration of identity resolution system and specific IIoT applications. Different applications have different service models and requirements on performance. Moreover, in some specific applications, it is not simply to choose which system to use, but to extract certain functions from systems, and then combine other new technologies to construct a specific solution. This article attempts to briefly discuss the potential identity resolution systems that may be used in IIoT. And we further present a general function-based reference framework to help researchers to assemble the advantages function of each system and apply them in specific IIoT scenarios. We hope our discussion can give some other researchers some inspiration and help in this field.

References

[1] "Cisco visual networking index: Forecast and trends, 2017–2022," Cisco, San Jose, CA, USA, Rep., 2018.

[2] F. Tao, Y. Zuo, L. D. Xu, and L. Zhang, "IoT-based intelligent perception and access of manufacturing resource toward cloud manufacturing," *IEEE Trans. Ind. Informat.*, vol. 10, no. 2, pp. 1547–1557, May 2014.

[3] L. D. Xu, W. He, and S. Li, "Internet of Things in industries: A survey," *IEEE Trans. Ind. Informat.*, vol. 10, no. 4, pp. 2233–2243, Nov. 2014.

[4] M. Hermann, T. Pentek, and B. Otto, "Design principles for industrie 4.0 scenarios," in *Proc. 49th Hawaii Int. Conf. Syst. Sci. (HICSS)*, Jan. 2016, pp. 3928–3937.

[5] E. Sisinni, A. Saifullah, S. Han, U. Jennehag, and M. Gidlund, "Industrial Internet of Things: Challenges, opportunities, and directions," *IEEE Trans. Ind. Informat.*, vol. 14, no. 11, pp. 4724–4734, Nov. 2018.

[6] J. Li, F. R. Yu, G. Deng, C. Luo, Z. Ming, and Q. Yan, "Industrial Internet: A survey on the enabling technologies, applications, and challenges," *IEEE Commun. Surveys Tuts.*, vol. 19, no. 3, pp. 1504–1526, 3rd Quart., 2017.

[7] R. Joshi, P. Didier, J. Jimenez, and T. Carey, *The Industrial Internet of Things Volume G5: Connectivity Framework*, Ind. Internet Consortium, Needham, MA, USA, 2017.

[8] M. R. Palattella *et al.*, "Internet of Things in the 5G era: Enablers, architecture, and business models," *IEEE J. Sel. Areas Commun.*, vol. 34, no. 3, pp. 510–527, Mar. 2016.

[9] S.-W. Lin *et al.*, *The Industrial Internet of Things Volume G1: Reference Architecture*, Ind. Internet Consortium, Needham, MA, USA, 2017, pp. 10–46.

[10] S. Malakuti *et al.*, *Digital Twins for Industrial Applications*, Ind. Internet Consortium, Needham, MA, USA, 2020.

[11] I. Halenar, M. Juhas, B. Juhasova, and D. Borkin, "Virtualization of production using digital twin technology," in *Proc. 20th Int. Carpathian Control Conf. (ICCC)*, 2019, pp. 1–5.

[12] "IEC 62890: Life-cycle management for systems and products used in industrial-process measurement, control and automation," Int. Electrotech. Commission, Geneva, Switzerland, Rep. IEC 62890, 2016.

[13] Z. Wang, T. Ye, and A. Xiong, "Research of food traceability technology based on the Internet of Things name service," in *Proc. IEEE Int. Conf. Internet Things (iThings) IEEE Green Comput. Commun. (GreenCom) IEEE Cyber Phys. Social Comput. (CPSCom) IEEE Smart Data (SmartData)*, 2016, pp. 100–106.

[14] G. Qiaolun and G. Tiegang, "Impacts of RFID/EPC on optimal decisions of reverse supply chain," in *Proc. Int. Conf. Bus. Comput. Global Inf.*, 2011, pp. 512–515.

[15] S. F. Wamba, L. A. Lefebvre, and E. Lefebvre, "Enabling intelligent B-to-B ecommerce supply chain management using RFID and the EPC network: A case study in the retail industry," in *Proc. 8th Int. Conf. Electron. Commerce New E-Commerce*, 2006, pp. 281–288.

[16] Q. Lin, H. Wang, X. Pei, and J. Wang, "Food safety traceability system based on blockchain and EPCIS," *IEEE Access*, vol. 7, pp. 20698–20707, 2019.

[17] P. Liu, W. Liu, Q. Li, M. Duan, Y. Wang, and Y. Dai, "A research on tracing code of culture of food safety traceability based on RFID and improved EPC," in *Proc. Int. Conf. Logist. Informat. Service Sci. (LISS)*, 2016, pp. 1–6.

[18] Z. X. liang and X. S. lian, "Application research of EPC network in reverse logistics," in *Proc. Int. Conf. Inf. Manag. Innov. Manag. Ind. Eng.*, vol. 3, 2012, pp. 369–372.

[19] R. Zheng and J. Qi, "EPC events in logistics process and the design of anti-counterfeiting patterns," in *Proc. Int. Conf. Inf. Technol. Comput. Eng. Manag. Sci.*, vol. 2, 2011, pp. 15–18.

[20] R. Oh and J. Park, "A development of active monitoring system for intelligent RFID logistics processing environment," in *Proc. Int. Conf. Adv. Lang. Process. Web Inf. Technol.*, 2008, pp. 358–361.

[21] S. Pal, M. Hitchens, and V. Varadharajan, "Modeling identity for the Internet of Things: Survey, classification and trends," in *Proc. 12th Int. Conf. Sens. Technol. (ICST)*, 2018, pp. 45–51.

[22] P. Mahalle, S. Babar, N. R. Prasad, and R. Prasad, "Identity management framework towards Internet of Things (IoT): Roadmap and key challenges," in *Proc. Int. Conf. Netw. Security Appl.*, 2010, pp. 430–439.

[23] K.-Y. Lam and C.-H. Chi, "Identity in the Internet-of-Things (IoT): New challenges and opportunities," in *Proc. Int. Conf. Inf. Commun. Security*, 2016, pp. 18–26.

[24] V. Gazis, "A survey of standards for machine-to-machine and the Internet of Things," *IEEE Commun. Surveys Tuts.*, vol. 19, no. 1, pp. 482–511, 1st Quart., 2017.

[25] A. Al-Fuqaha, M. Guizani, M. Mohammadi, M. Aledhari, and M. Ayyash, "Internet of Things: A survey on enabling technologies, protocols, and applications," *IEEE Commun. Surveys Tuts.*, vol. 17, no. 4, pp. 2347–2376, 4th Quart., 2015.

[26] X. Zhu and Y. Badr, "A survey on blockchain-based identity management systems for the Internet of Things," in *Proc. IEEE Int. Conf. Internet Things (iThings) IEEE Green Comput. Commun. (GreenCom) IEEE Cyber Phys. Soc. Comput. (CPSCom) IEEE Smart Data (SmartData)*, 2018, pp. 1568–1573.

[27] Y. Hsu, J. Chiu, and J. S. Liu, "Digital twins for industry 4.0 and beyond," in *Proc. IEEE Int. Conf. Ind. Eng. Eng. Manag. (IEEM)*, 2019, pp. 526–530.

[28] J. Ríos, J. C. Hernandez, M. Oliva, and F. Mas, "Product avatar as digital counterpart of a physical individual product: Literature review and implications in an aircraft," in *Proc. ISPE CE*, 2015, pp. 657–666.

[29] *Batch Control—Part 1: Models and Terminology*, IEC Standard 61512-1, 1997.

[30] B. Wally, C. Huemer, A. Mazak, and M. Wimmer, "IEC 62264-2 for automationML," in *Proc. 5th Autom. ML User Conf.*, 2018, pp. 1–7.

[31] *Engineering Data Exchange Format for Use in Industrial Automation Systems Engineering—Automation Markup Language—Part 1: Architecture and General Requirements*, IEC Standard 62714-1, 2014.

[32] *Representation of Process Control Engineering—Requests in P&I Diagrams and Data Exchange Between P&ID Tools and PCE-CAE Tools*, IEC Standard 62424, 2008.

[33] K. M. Alam and A. El Saddik, "C2PS: A digital twin architecture reference model for the cloud-based cyber-physical systems," *IEEE Access*, vol. 5, pp. 2050–2062, 2017.

[34] A. Canedo, "Industrial IoT lifecycle via digital twins," in *Proc. Int. Conf. Hardw. Softw. Codesign Syst. Synth. (CODES+ISSS)*, 2016, p. 29.

[35] H. Shen, Q. Zhou, J. Zhao, and X. Liu, "A novel product life-cycle management architecture of construction machinery," in *Proc. IEEE Int. Conf. Mechatron. Autom. (ICMA)*, 2017, pp. 899–903.

[36] X. Yusen, N. F. Bondaletova, V. I. Kovalev, and A. V. Komrakov, "Digital twin concept in managing industrial capital construction projects life cycle," in *Proc. 11th Int. Conf. Manag. Large-Scale Syst. Develop. (MLSD)*, 2018, pp. 1–3.

[37] J. Vachálek *et al.*, "The digital twin of an industrial production line within the industry 4.0 concept," in *Proc. 21st Int. Conf. Process Control (PC)*, 2017, pp. 258–262.

[38] P. Zehnder and D. Riemer, "Representing industrial data streams in digital twins using semantic labeling," in *Proc. IEEE Int. Conf. Big Data (Big Data)*, 2018, pp. 4223–4226.

[39] C. Esposito, A. Castiglione, F. Palmieri, and A. D. Santis, "Integrity for an event notification within the industrial Internet of Things by using group signatures," *IEEE Trans. Ind. Informat.*, vol. 14, no. 8, pp. 3669–3678, Aug. 2018.

[40] S. Yu, M. Liu, W. Dou, X. Liu, and S. Zhou, "Networking for big data: A survey," *IEEE Commun. Surveys Tuts.*, vol. 19, no. 1, pp. 531–549, 1st Quart., 2017.

[41] D. van Bokkem, R. Hageman, G. Koning, L. Nguyen, and N. Zarin, "Self-sovereign identity solutions: The necessity of blockchain technology," 2019. [Online]. Available: arXiv:1904.12816.

[42] X. Yu, Y. Zhang, X. Chen, and Y. Zu, "The product life cycle design in plant extracts industry," in *Proc. IEEE 18th Int. Conf. Ind. Eng. Eng. Manag.*, 2011, pp. 573–575.

[43] X. Jiang and C. Ma, "Concerning the important position of product modeling design in the product life cycle strategy," in *Proc. IEEE 18th Int. Conf. Ind. Eng. Eng. Manag.*, 2011, pp. 540–542.

[44] S. Shrestha, D. S. Kim, S. Lee, and J. S. Park, "A peer-to-peer RFID resolution framework for supply chain network," in *Proc. 2nd Int. Conf. Future Netw.*, 2010, pp. 318–322.

[45] L. Barreto, A. Amaral, and T. Pereira, "Industry 4.0 implications in logistics: An overview," *Procedia Manuf.*, vol. 13, pp. 1245–1252, 2017.

[46] P. Autenrieth, C. Lärcher, C. Pfeiffer, T. Winkens, and L. Martin, "Current significance of IT-infrastructure enabling industry 4.0 in large companies," in *Proc. IEEE Int. Conf. Eng. Technol. Innov. (ICE/ITMC)*, 2018, pp. 1–8.

[47] T. Qiu *et al.*, "Edge computing in industrial Internet of Things: Architecture, advances and challenges," *IEEE Commun. Surveys Tuts.*, vol. 22, no. 4, pp. 2462–2488, 4th Quart., 2020.

[48] S. Torabi, A. Boukhtouta, C. Assi, and M. Debbabi, "Detecting Internet abuse by analyzing passive DNS traffic: A survey of implemented systems," *IEEE Commun. Surveys Tuts.*, vol. 20, no. 4, pp. 3389–3415, 4th Quart., 2018.

[49] S. Yu, S. Guo, and I. Stojmenovic, "Fool me if you can: Mimicking attacks and anti-attacks in cyberspace," *IEEE Trans. Comput.*, vol. 64, no. 1, pp. 139–151, Jan. 2015.

[50] S. Yu, W. Zhou, R. Doss, and W. Jia, "Traceback of DDoS attacks using entropy variations," *IEEE Trans. Parallel Distrib. Syst.*, vol. 22, no. 3, pp. 412–425, Mar. 2011.

[51] B. Fabian and O. Günther, "Security challenges of the EPCglobal network," *Commun. ACM*, vol. 52, no. 7, pp. 121–125, 2009.

[52] X. Li, Y. Liu, Y. Tian, N. Kong, Y. Wang, and W. Mao, "Towards an equitable federated name service for the Internet of Things," in *Proc. IEEE Int. Conf. Green Comput. Commun. IEEE Internet Things IEEE Cyber Phys. Soc. Comput.*, 2013, pp. 1109–1115.

[53] K. Tange, M. De Donno, X. Fafoutis, and N. Dragoni, "A systematic survey of industrial Internet of Things security: Requirements and fog computing opportunities," *IEEE Commun. Surveys Tuts.*, vol. 22, no. 4, pp. 2489–2520, 1st Quart., 2020.

[54] G. Fersi, W. Louati, and M. B. Jemaa, "Distributed hash table-based routing and data management in wireless sensor networks: A survey," *Wireless Netw.*, vol. 19, no. 2, pp. 219–236, 2013.

[55] S. Nakamoto. (2008). *Bitcoin: A Peer-to-Peer Electronic Cash System*. [Online]. Available: https://bitcoin.org/bitcoin.pdf

[56] (Jun. 2010). *An Introduction to Petname Systems*. [Online]. Available: https://sawtooth.hyperledger.org/docs/core/releases/1.0/introduction.html

[57] B. Feng, H. Zhang, H. Zhou, and S. Yu, "Locator/identifier split networking: A promising future Internet architecture," *IEEE Commun. Surveys Tuts.*, vol. 19, no. 4, pp. 2927–2948, 4th Quart., 2017.

[58] M. Schanzenbach, C. Banse, and J. Schütte, "Practical decentralized attribute-based delegation using secure name systems," in *Proc. 17th IEEE Int. Conf. Trust Security Privacy Comput. Commun. 12th IEEE Int. Conf. Big Data Sci. Eng. (TrustCom/BigDataSE)*, 2018, pp. 244–251.

[59] J. Xie *et al.*, "A survey of blockchain technology applied to smart cities: Research issues and challenges," *IEEE Commun. Surveys Tuts.*, vol. 21, no. 3, pp. 2794–2830, 3rd Quart., 2019.

[60] Y. Fu, F. R. Yu, C. Li, T. H. Luan, and Y. Zhang, "Vehicular blockchain-based collective learning for connected and autonomous vehicles," *IEEE Wireless Commun.*, vol. 27, no. 2, pp. 197–203, Apr. 2020.

[61] R. Yang, F. R. Yu, P. Si, Z. Yang, and Y. Zhang, "Integrated blockchain and edge computing systems: A survey, some research issues and challenges," *IEEE Commun. Surveys Tuts.*, vol. 21, no. 2, pp. 1508–1532, 2nd Quart., 2019.

[62] S. Arshad, M. A. Azam, M. H. Rehmani, and J. Loo, "Recent advances in information-centric networking-based Internet of Things (ICN-IoT)," *IEEE Internet Things J.*, vol. 6, no. 2, pp. 2128–2158, Apr. 2019.

[63] L. Zhang *et al.*, "Named data networking," *ACM SIGCOMM Comput. Commun. Rev.*, vol. 44, no. 3, pp. 66–73, 2014.

[64] O. Wannenwetsch and T. A. Majchrzak, "On constructing persistent identifiers with persistent resolution targets," in *Proc. Feder. Conf. Comput. Sci. Inf. Syst. (FedCSIS)*, Sep. 2016, pp. 1031–1040.

[65] O. Schmitt, T. A. Majchrzak, and S. Bingert, "Experimental realization of a persistent identifier infrastructure stack for named data networking," in *Proc. IEEE Int. Conf. Netw. Architect. Storage (NAS)*, Aug. 2015, pp. 33–38.

[66] A. Karakannas and Z. Zhao, *Information Centric Networking for Delivering Big Data With Persistent Identifiers*, Univ. Amsterdam, Amsterdam, The Netherlands, 2014.

[67] D. L. Brock, "The electronic product code (EPC)," Auto-ID Center, Massachusetts Inst. Technol., Cambridge, MA, USA, White Paper, pp. 1–21, 2001.

[68] O. G. Sobrinho and C. E. Cugnasca, "An overview of the EPCglobal network," *IEEE Latin America Trans.*, vol. 11, no. 4, pp. 1053–1059, Jun. 2013.

[69] M. Mealling, "Auto-ID object name service (ONS) 1.0," Auto-ID Center, Massachusetts Inst. Technol., Cambridge, MA, USA, Working Draft, 2003.

[70] D. Engels, "The use of the electronic product code," Auto-ID Center, Massachusetts Inst. Technol., Cambridge, MA, USA, Rep., 2003.

[71] D. Brock and C. Cummins, "EPC tag data specification," Auto-ID Center, Massachusetts Inst. Technol., Cambridge, MA, USA, White Paper, 2003.

[72] C. Rinderknecht, "An algorithm for validating ASN.1 (x.680) specifications using set constraints," *Comput. J.*, vol. 46, no. 4, pp. 401–420, 2003.

[73] M. Mealling, "A URN namespace of object identifiers," IETF, RFC 3061, Feb. 2001. [Online]. Available: http://www.ietf.org/rfc/rfc3061

[74] *Information Technology—Open Systems Interconnection—Part 1: Object Identifier Resolution System*. Accessed: Sep. 2011. [Online]. Available: https://www.iso.org/standard/45247.html

[75] *Information Technology—Open Systems Interconnection—Part 2: Procedures for the Object Identifier Resolution System Operational Agency*. Accessed: Sep. 2011. [Online]. Available: https://www.iso.org/standard/56543.html

[76] S. Sun, L. Lannom, and B. Boesch, "Handle system overview," IETF, RFC 3650, 2003.

[77] S. Sun, S. Reilly, and L. Lannom, "Handle system namespace and service definition," IETF, RFC 3651, 2003.

[78] S. Sun, S. Reilly, L. Lannom, and J. Petrone, "Handle system protocol (ver 2.1) specification," IETF, RFC 3652, 2003.

[79] N. Koshizuka and K. Sakamura, "Ubiquitous ID: Standards for ubiquitous computing and the Internet of Things," *IEEE Pervasive Comput.*, vol. 9, no. 4, pp. 98–101, Oct.–Dec. 2010.

[80] K. Amanuma, "Product identification system based on Ubiquitous ID technology," in *Proc. 8th IEEE Int. Conf. Ind. Informat.*, Jul. 2010, pp. 406–411.

[81] *T-Engine Forum*, Ubiquitous ID Center, Tokyo, Japan, 2006.

[82] *Simplified Ucode Resolution Protocol*, Ubiquitous ID Center, Tokyo, Japan, 2008.

[83] *Ubiquitous Code: Ucode*, Ubiquitous ID Center, Tokyo, Japan, 2009.

[84] M. Wachs, M. Schanzenbach, and C. Grothoff, "A censorship-resistant, privacy-enhancing and fully decentralized name system," in *Proc. Int. Conf. Cryptol. Netw. Security*, 2014, pp. 127–142.

[85] M. Schanzenbach, G. Bramm, and J. Schütte, "ReclaiMID: Secure, self-sovereign identities using name systems and attribute-based encryption," in *Proc. 17th IEEE Int. Conf. Trust Security Privacy Comput. Commun. 12th IEEE Int. Conf. Big Data Sci. Eng. (TrustCom/BigDataSE)*, Aug. 2018, pp. 946–957.

[86] (2017). *NSA's MORECOWBELL: Knell for DNS*. [Online]. Available: http://goodtimesweb.org/surveillance/2015/MORECOWBELL-Analysis-Grothoff-etal.pdf

[87] M. Ali, J. Nelson, R. Shea, and M. J. Freedman, "BlockStack: A global naming and storage system secured by blockchains," in *Proc. {USENIX} Annu. Tech. Conf. ({USENIX}{ATC})*, 2016, pp. 181–194.

[88] J. Nelson, M. Ali, R. Shea, and M. J. Freedman, "Extending existing blockchains with virtualchain," in *Proc. Workshop Distrib. Cryptocurrencies Consensus Ledgers*, 2016, pp. 1–5.

[89] M. Ali, J. Nelson, R. Shea, and M. J. Freedman, "Bootstrapping trust in distributed systems with blockchains," *USENIX Mag.*, vol. 41, no. 3, pp. 1–7, 2016.

[90] (May 2019). *Blockstack Technical Whitepaper v 2.0*. [Online]. Available: https://uploadsssl.webflow.com/5e7b1a27d160ce49af1c24e1/5f1596b27c92eb866da76462_whitepaper.pdf

[91] (Oct. 2017). *Blockstack Token Whitepaper*. [Online]. Available: https://www.sec.gov/Archives/edgar/data/1719379/000110465919020748/a18-15736_1ex1a15addexhbd2.htm

[92] D. Reed, M. Sporny, D. Longley, C. Allen, R. Grant, and M. Sabadello. *Decentralized Identifiers (DIDs) V1.0 Core Architecture, Data Model, and Representations*. [Online]. Available: https://w3c.github.io/did-core/

[93] O. Kolkman and R. Gieben, "DNSSEC operational practices," IETF, Rep. RFC 4641, Sep. 2006.

[94] Y. An, F. R. Yu, J. Li, J. Chen, and V. C. M. Leung, "Edge intelligence (EI)-enabled HTTP anomaly detection framework for the Internet of Things (IoT)," *IEEE Internet Things J.*, early access, Sep. 18, 2020, doi: 10.1109/JIOT.2020.3024645.

[95] S. Ariyapperuma and C. J. Mitchell, "Security vulnerabilities in DNS and DNSSEC," in *Proc. 2nd Int. Conf. Availability Rel. Security (ARES)*, 2007, pp. 335–342.

[96] S. Yu, Y. Tian, S. Guo, and D. O. Wu, "Can we beat DDoS attacks in clouds?" *IEEE Trans. Parallel Distrib. Syst.*, vol. 25, no. 9, pp. 2245–2254, Sep. 2014.

[97] Q. Yan, W. Huang, X. Luo, Q. Gong, and F. R. Yu, "A multi-level DDoS mitigation framework for the industrial Internet of Things," *IEEE Commun. Mag.*, vol. 56, no. 2, pp. 30–36, Feb. 2018.

[98] (Oct. 2016). *Overview of the Digital Object Architecture (DOA)*. [Online]. Available: https://www.internetsociety.org/resources/doc/2016/overview-of-the-digital-objectarchitecture-doa/

[99] S. Yu, W. Zhou, W. Jia, S. Guo, Y. Xiang, and F. Tang, "Discriminating DDoS attacks from flash crowds using flow correlation coefficient," *IEEE Trans. Parallel Distrib. Syst.*, vol. 23, no. 6, pp. 1073–1080, Jun. 2012.

[100] H. Huang, X. Zhou, and J. Liu, "Food supply chain traceability scheme based on blockchain and EPC technology," in *Proc. Int. Conf. Smart Blockchain*, 2019, pp. 32–42.

[101] P. S. Kim, "Comparison and analysis of DNS and DOA for Internet of Things naming system," in *Proc. Int. Conf. Artif. Intell. Inf. Commun. (ICAIIC)*, 2019, pp. 552–556.

[102] S. Bano *et al.*, "Consensus in the age of blockchains," 2017. [Online]. Available: arXiv:1711.03936.

[103] A. Gómez-Cárdenas, X. Masip-Bruin, E. Marín-Tordera, S. Kahvazadeh, and J. Garcia, "A hash-based naming strategy for the fog-to-cloud computing paradigm," in *Proc. Eur. Conf. Parallel Process.*, 2017, pp. 316–324.

[104] Z. Yan, N. Kong, Y. Tian, and Y. Park, "A universal object name resolution scheme for IoT," in *Proc. IEEE Int. Conf. Green Comput. Commun. IEEE Internet Things IEEE Cyber Phys. Soc. Comput.*, 2013, pp. 1120–1124.

[105] W. Ben Rhaiem, W. Louati, and D. Zeghlache, "mhDHT: A scalable DHT-based name resolution system for the future Internet," in *Proc. 3rd Int. Conf. Netw. Future (NOF)*, Nov. 2012, pp. 1–5.

[106] F. Berber, P. Wieder, and R. Yahyapour, "A high-performance persistent identification concept," in *Proc. IEEE Int. Conf. Netw. Architect. Storage (NAS)*, 2016, pp. 1–10.

[107] J. Jiang, S. Wen, S. Yu, Y. Xiang, and W. Zhou, "Identifying propagation sources in networks: State-of-the-art and comparative studies," *IEEE Commun. Surveys Tuts.*, vol. 19, no. 1, pp. 465–481, 1st Quart., 2017.

[108] Z. Hu, L. Zhu, J. Heidemann, A. Mankin, D. Wessels, and P. Hoffman, "Specification for DNS over transport layer security (TLS)," IETF, RFC 7858, 2016.

[109] S. Dickinson, D. Gillmor, and T. Reddy, *Usage Profiles for DNS Over TLS and DNS Over DTLS*, Internet Eng. Task Force, Fremont, CA, USA, 2018.

[110] P. Hoffman and P. McManus, "DNS queries over HTTPS (DoH)," IETF, RFC 8484, 2018.

[111] R. Cox, A. Muthitacharoen, and R. T. Morris, "Serving DNS using a peer-to-peer lookup service," in *Proc. Int. Workshop Peer-to-Peer Syst.*, 2002, pp. 155–165.

[112] B. Fabian and O. Gunther, "Distributed ONS and its impact on privacy," in *Proc. IEEE Int. Conf. Commun.*, Jun. 2007, pp. 1223–1228.

[113] Y. Doi, S. Wakayama, M. Ishiyama, S. Ozaki, and A. Inoue, "On scalability of DHT-DNS hybrid naming system," in *Proc. Asian Internet Eng. Conf.*, Nov. 2006.

[114] H. Seike, T. Hamada, T. Sumitomo, and N. Koshizuka, "Blockchain-based ubiquitous code ownership management system without hierarchical structure," in *Proc. IEEE SmartWorld Ubiquitous Intell. Comput. Adv. Trusted Comput. Scalable Comput. Commun. Cloud Big Data Comput. Internet People Smart City Innov. (SmartWorld/SCALCOM/UIC/ATC/CBDCom/IOP/SCI)*, Oct. 2018, pp. 271–276.

[115] E. Jung, Y. Choi, J. S. Lee, and H. J. Kim, "An OID-based identifier framework supporting the interoperability of heterogeneous identifiers," in *Proc. 14th Int. Conf. Adv. Commun. Technol. (ICACT)*, Feb. 2012, pp. 304–308.

[116] *OID-Based Resolution Framework for Heterogeneous Identifiers and Locators*, ITU-T, Geneva, Switzerland, 2015.

[117] T. Yashiro, S. Kobayashi, N. Koshizuka, and K. Sakamura, "An Internet of Things (IoT) architecture for embedded appliances," in *Proc. IEEE Region 10 Humanitarian Technol. Conf.*, Aug. 2013, pp. 314–319.

[118] C. Wagner *et al.*, "The role of the industry 4.0 asset administration shell and the digital twin during the life cycle of a plant," in *Proc. 22nd IEEE Int. Conf. Emerg. Technol. Factory Autom. (ETFA)*, 2017, pp. 1–8.

[119] M. Al-Bahri, A. Yankovsky, R. Kirichek, and A. Borodin, "Smart system based on DOA & IoT for products monitoring & anti-counterfeiting," in *Proc. 4th MEC Int. Conf. Big Data Smart City (ICBDSC)*, 2019, pp. 1–5.

[120] *Object Identifier-Based Resolution Framework for IoT Grouped Services*, ITU-T, Geneva, Switzerland, 2018.

[121] A. Abraham. (2017). *Self-Sovereign Identity*. [Online]. Available: http://egiz.gv.at/

[122] R. Soltani, U. T. Nguyen, and A. An, "Practical key recovery model for self-sovereign identity based digital wallets," in *Proc. IEEE Int. Conf. Depend. Auton. Secure Comput. Int. Conf. Pervasive Intell. Comput. Int. Conf. Cloud Big Data Comput. Int. Conf. Cyber Sci. Technol. Congr. (DASC/PiCom/CBDCom/CyberSciTech)*, 2019, pp. 320–325.

[123] (Apr. 2016). *The Path to Self-Sovereign Identity*. [Online]. Available: http://www.lifewithalacrity.com/2016/04/the-path-to-self-soverereign-identity.html

[124] M. S. Ferdous, F. Chowdhury, and M. O. Alassafi, "In search of self-sovereign identity leveraging blockchain technology," *IEEE Access*, vol. 7, pp. 103059–103079, 2019.

[125] N. Naik and P. Jenkins, "Self-sovereign identity specifications: Govern your identity through your digital wallet using blockchain technology," in *Proc. 8th IEEE Int. Conf. Mobile Cloud Comput. Services Eng. (MobileCloud)*, 2020, pp. 90–95.

[126] Y. Liu, Q. Lu, H. Y. Paik, X. Xu, S. Chen, and L. Zhu, "Design pattern as a service for blockchain-based self-sovereign identity," *IEEE Softw.*, vol. 37, no. 5, pp. 30–36, Sep./Oct. 2020.

[127] Q. Stokkink and J. Pouwelse, "Deployment of a blockchain-based self-sovereign identity," in *Proc. IEEE Int. Conf. Internet Things (iThings) IEEE Green Comput. Commun. (GreenCom) IEEE Cyber Phys. Soc. Comput. (CPSCom) IEEE Smart Data (SmartData)*, 2018, pp. 1336–1342.

[128] B. Houtan, A. S. Hafid, and D. Makrakis, "A survey on blockchain-based self-sovereign patient identity in healthcare," *IEEE Access*, vol. 8, pp. 90478–90494, 2020.

[129] C. Liang, F. R. Yu, and X. Zhang, "Information-centric network function virtualization over 5G mobile wireless networks," *IEEE Netw.*, vol. 29, no. 3, pp. 68–74, May/Jun. 2015.

[130] C. Fang, F. R. Yu, T. Huang, J. Liu, and Y. Liu, "A survey of green information-centric networking: Research issues and challenges," *IEEE Commun. Surveys Tuts.*, vol. 17, no. 3, pp. 1455–1472, 3rd Quart., 2015.

[131] Y. Zhou, F. R. Yu, J. Chen, and Y. Kuo, "Resource allocation for information-centric virtualized heterogeneous networks with in-network caching and mobile edge computing," *IEEE Trans. Veh. Technol.*, vol. 66, no. 12, pp. 11339–11351, Dec. 2017.

[132] V. Jacobson, D. K. Smetters, J. D. Thornton, M. F. Plass, N. H. Briggs, and R. L. Braynard, "Networking named content," in *Proc. 5th Int. Conf. Emerg. Netw. Exp. Technol.*, 2009, pp. 1–12.

[133] D. Trossen and G. Parisis, "Designing and realizing an information-centric Internet," *IEEE Commun. Mag.*, vol. 50, no. 7, pp. 60–67, Jul. 2012.

[134] D. Raychaudhuri, K. Nagaraja, and A. Venkataramani, "MobilityFirst: A robust and trustworthy mobility-centric architecture for the future Internet," *ACM SIGMOBILE Mobile Comput. Commun. Rev.*, vol. 16, no. 3, pp. 2–13, 2012.

[135] G. Xylomenos *et al.*, "A survey of information-centric networking research," *IEEE Commun. Surveys Tuts.*, vol. 16, no. 2, pp. 1024–1049, 2nd Quart., 2014.

[136] T. Vu *et al.*, "DMap: A shared hosting scheme for dynamic identifier to locator mappings in the global Internet," in *Proc. IEEE 32nd Int. Conf. Distrib. Comput. Syst.*, 2012, pp. 698–707.

[137] S. C. Nelson, G. Bhanage, and D. Raychaudhuri, "GSTAR: Generalized storage-aware routing for mobilityfirst in the future mobile Internet," in *Proc. 6th Int. Workshop MobiArch*, 2011, pp. 19–24.

[138] X. Liu, Z. Wang, C. Jin, F. Li, and G. Li, "A blockchain-based medical data sharing and protection scheme," *IEEE Access*, vol. 7, pp. 118943–118953, 2019.

[139] H. Lee, K. Sung, K. Lee, J. Lee, and S. Min, "Economic analysis of blockchain technology on digital platform market," in *Proc. IEEE 23rd Pac. Rim Int. Symp. Depend. Comput. (PRDC)*, Dec. 2018, pp. 94–103.

[140] C. Qiu, F. R. Yu, H. Yao, C. Jiang, F. Xu, and C. Zhao, "Blockchain-based software-defined industrial Internet of Things: A dueling deep Q -learning approach," *IEEE Internet Things J.*, vol. 6, no. 3, pp. 4627–4639, Jun. 2019.

[141] R. Zhang, F. R. Yu, J. Liu, T. Huang, and Y. Liu, "Deep reinforcement learning (DRL)-based device-to-device (D2D) caching with blockchain and mobile edge computing," *IEEE Trans. Wireless Commun.*, vol. 19, no. 10, pp. 6469–6485, Oct. 2020.

[142] R. Zhang, F. R. Yu, J. Liu, R. Xie, and T. Huang, "Blockchain-incentivized D2D and mobile edge caching: A deep reinforcement learning approach," *IEEE Netw.*, vol. 34, no. 4, pp. 150–157, Jul./Aug. 2020.

[143] J. Liu, B. Li, L. Chen, M. Hou, F. Xiang, and P. Wang, "A data storage method based on blockchain for decentralization DNS," in *Proc. IEEE 3rd Int. Conf. Data Sci. Cybersp. (DSC)*, Jun. 2018, pp. 189–196.

[144] X. Duan, Z. Yan, G. Geng, and B. Yan, "DNSLedger: Decentralized and distributed name resolution for ubiquitous IoT," in *Proc. IEEE Int. Conf. Consum. Electron. (ICCE)*, Jan. 2018, pp. 1–3.

[145] X. Wang, K. Li, H. Li, Y. Li, and Z. Liang, "ConsortiumDNS: A distributed domain name service based on consortium chain," in *Proc. IEEE 19th Int. Conf. High Perform. Comput. Commun. IEEE 15th Int. Conf. Smart City IEEE 3rd Int. Conf. Data Sci. Syst. (HPCC/SmartCity/DSS)*, 2017, pp. 617–620.

[146] Z. Yu, D. Xue, J. Fan, and C. Guo, "DNSTSM: DNS cache resources trusted sharing model based on consortium blockchain," *IEEE Access*, vol. 8, pp. 13640–13650, 2020.

[147] W. Yoon, J. Im, I. Choi, and D. Kim, "Blockchain-based object name service with tokenized authority," *IEEE Trans. Services Comput.*, vol. 13, no. 2, pp. 329–342, Mar./Apr. 2020.

[148] W. Yoon, I. Choi, and D. Kim, "BlockONS: blockchain based object name service," in *Proc. IEEE Int. Conf. Blockchain Cryptocurrency (ICBC)*, May 2019, pp. 219–226.

[149] S. Wang, S. Zhu, and Y. Zhang, "Blockchain-based mutual authentication security protocol for distributed RFID systems," in *Proc. IEEE Symp. Comput. Commun. (ISCC)*, Jun. 2018, pp. 74–77.

[150] Y. Liu, F. R. Yu, X. Li, H. Ji, and V. C. M. Leung, "Blockchain and machine learning for communication and networking systems," *IEEE Commun. Surveys Tuts.*, vol. 22, no. 2, pp. 1392–1431, 2nd Quart., 2020.

[151] Y. Jin, M. Tomoishi, and S. Matsuura, "A detection method against DNS cache poisoning attacks using machine learning techniques: Work in progress," in *Proc. IEEE 18th Int. Symp. Netw. Comput. Appl. (NCA)*, Sep. 2019, pp. 1–3.

[152] M. S. Khan, K. Ferens, and W. Kinsner, "A cognitive multifractal approach to characterize complexity of non-stationary and malicious DNS data traffic using adaptive sliding window," in *Proc. IEEE 14th Int. Conf. Cogn. Informat. Cogn. Comput. (ICCI*CC)*, Jul. 2015, pp. 76–83.

[153] S. Tian, C. Fang, J. Liu, and Z. Lei, "Detecting malicious domains by massive DNS traffic data analysis," in *Proc. 8th Int. Conf. Intell. Human–Mach. Syst. Cybern. (IHMSC)*, vol. 1, Aug. 2016, pp. 130–133.

[154] J. Jin, Z. Yan, G. Geng, and B. Yan, "Botnet domain name detection based on machine learning," in *Proc. 6th Int. Conf. Wireless Mobile Multi Media (ICWMMN)*, Nov. 2015, pp. 273–276.

[155] H. Tran, A. Nguyen, P. Vo, and T. Vu, "DNS graph mining for malicious domain detection," in *Proc. IEEE Int. Conf. Big Data (Big Data)*, Dec. 2017, pp. 4680–4685.

[156] A. Moubayed, M. Injadat, A. Shami, and H. Lutfiyya, "DNS typo-squatting domain detection: A data analytics &; machine learning based approach," in *Proc. IEEE Global Commun. Conf. (GLOBECOM)*, Dec. 2018, pp. 1–7.

[157] C. Liu, L. Dai, W. Cui, and T. Lin, "A byte-level CNN method to detect DNS tunnels," in *Proc. IEEE 38th Int. Perform. Comput. Commun. Conf. (IPCCC)*, Oct. 2019, pp. 1–8.

[158] D. B. Rawat and S. R. Reddy, "Software defined networking architecture, security and energy efficiency: A survey," *IEEE Commun. Surveys Tuts.*, vol. 19, no. 1, pp. 325–346, 1st Quart., 2017.

[159] P. Gonçalves, J. Ferreira, P. Pedreiras, and D. Corujo, "Adapting SDN datacenters to support cloud IIoT applications," in *Proc. IEEE 20th Conf. Emerg. Technol. Factory Autom. (ETFA)*, Sep. 2015, pp. 1–4.

[160] X. Li, D. Li, J. Wan, C. Liu, and M. Imran, "Adaptive transmission optimization in SDN-based industrial Internet of Things with edge computing," *IEEE Internet Things J.*, vol. 5, no. 3, pp. 1351–1360, Jun. 2018.

[161] M. Du and K. Wang, "An SDN-enabled pseudo-honeypot strategy for distributed denial of service attacks in industrial Internet of Things," *IEEE Trans. Ind. Informat.*, vol. 16, no. 1, pp. 648–657, Jan. 2020.

[162] R. Chaudhary, G. S. Aujla, S. Garg, N. Kumar, and J. J. P. C. Rodrigues, "SDN-enabled multi-attribute-based secure communication for smart grid in IIoT environment," *IEEE Trans. Ind. Informat.*, vol. 14, no. 6, pp. 2629–2640, Jun. 2018.

[163] S. Al-Rubaye, E. Kadhum, Q. Ni, and A. Anpalagan, "Industrial Internet of Things driven by SDN platform for smart grid resiliency," *IEEE Internet Things J.*, vol. 6, no. 1, pp. 267–277, Feb. 2019.

[164] Q. Yan, F. R. Yu, Q. Gong, and J. Li, "Software-defined networking (SDN) and distributed denial of service (DDoS) attacks in cloud computing environments: A survey, some research issues, and challenges," *IEEE Commun. Surveys Tuts.*, vol. 18, no. 1, pp. 602–622, 1st Quart., 2016.

[165] X. Xing, T. Luo, J. Li, and Y. Hu, "A defense mechanism against the DNS amplification attack in SDN," in *Proc. IEEE Int. Conf. Netw. Infrastruct. Digit. Content (IC-NIDC)*, Sep. 2016, pp. 28–33.

[166] N. Sahri and K. Okamura, "Collaborative spoofing detection and mitigation—SDN based looping authentication for DNS services," in *Proc. IEEE 40th Annu. Comput. Softw. Appl. Conf. (COMPSAC)*, vol. 2, Jun. 2016, pp. 565–570.

[167] Y. Al Mtawa, A. Memari, A. Haque, and H. Lutfiyya, "Evaluating QoS in SDN-based EPC: A comparative analysis," in *Proc. 15th Int. Wireless Commun. Mobile Comput. Conf. (IWCMC)*, Jun. 2019, pp. 1279–1286.

[168] Y.-h. Kim, H.-k. Lim, K.-H. Kim, and Y.-H. Han, "A SDN-based distributed mobility management in LTE/EPC network," *J. Supercomput.*, vol. 73, no. 7, pp. 2919–2933, 2017.