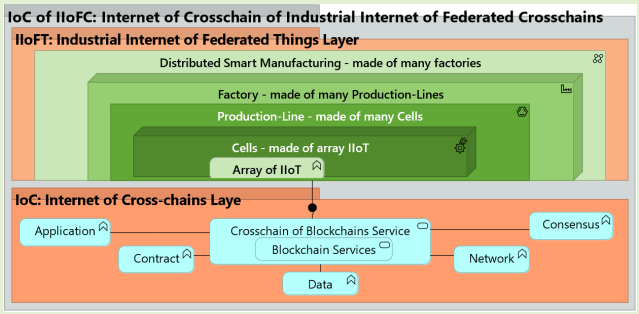
QoS-Aware Federated Crosschain-Based Model-Driven Reference Architecture for IIoT Sensor Networks in Distributed Manufacturing

Akila Siriweera, *Member, IEEE*, and Keitaro Naruse, *Member, IEEE*

***Abstract*—The realm of the Industrial Internet of Things (IIoT) encompasses a broad spectrum of sensors that are integral to distributed smart manufacturing (DSM). The miscellaneous IIoT sensors deployed for DSM are distributed and operate in a hierarchical and federated structure. Nonetheless, fulfilling essential quality of service (QoS) requirements, such as ensuring security and privacy (integrity) while maintaining scalability and interoperability (robustness), poses a profound challenge for the DSM cloud service platform. Although blockchain technologies have been used to safeguard integrity, the first two generations have imposed constraints on robustness. In contrast, the third-generation blockchain, a.k.a. decentralized crosschain ecosystem, complements Web 3.0 and metaverse and can mitigate the constraints of previous generations. Moreover, blockchain-based ad hoc solutions for DSM use cases are abundant; they often suffer from limited adaptability and unique or homogeneous use cases from a software engineering perspective. A holistic architectural modeling process (AMP) leading to a software reference architecture (SRA) is preferred when alleviating ad hoc constraints. Therefore, we proposed an AMP for SRA for crosschain-based DSM that safeguards integrity while preserving robustness. In the work described in this article, we have conducted the following system—modeling process. First, we propose a novel software AMP for DSM. Second, we deduce a modeled SRA based on a crosschain. Third, we infer a modeled system architecture (SA). Empirical experiment results demonstrate that our proposed crosschain-based method outperforms the widely used on-chain-based method while achieving our objectives efficiently.**



***Index Terms*—Distributed smart manufacturing (DSM), Industrial Internet of Things (IIoT), industrial metaverse, software modeling, software reference architecture (SRA), Web 30.**

Nomenclature

AMP	Architecture modeling process.
DMC	Decentralized multicloud.
DSM	Distributed smart manufacturing.
IIoFC	Industrial internet of federated crosschain.
IoC	Internet of crosschain.
MSM	Multisignature shared mempool.
SA	System architecture.
SMF	Smart manufacturing factory/facility.
SRA	Software reference architecture.

Manuscript received 18 September 2023; revised 14 October 2023; accepted 14 October 2023. Date of publication 30 October 2023; date of current version 30 November 2023. The associate editor coordinating the review of this article and approving it for publication was Prof. Huang Chen Lee. *(Corresponding author: Akila Siriweera.)*

The authors are with the School of Computer Science and Engineering, The University of Aizu, Aizuwakamatsu 965-8580, Japan (e-mail: sanjaya@u-aizu.ac.jp; naruse@u-aizu.ac.jp).

Digital Object Identifier 10.1109/JSEN.2023.3325342

I. Introduction

THE global Industrial Internet of Things (IIoT) accounts for more than 10% of the global Internet of Things (IoT) market share, and the demand for IIoT is expected to reach U.S. $110 billion by 2025 [1], [2], [3]. Moreover, IIoT plays a crucial role in smart manufacturing, an integral objective of Industry 4.0 [4], [5], [6]. Smart manufacturing involves factory automation systems (FAS), supply chain management (SCM), cyber-physical systems (CPS), and metaverse, which rely heavily on IIoT and associated entities, such as the Internet of Robotic Things and robotics generally [5], [6], [7].

Modern SCM processes are fast-moving and distributed globally. Smart manufacturing factories in FAS, SCM, CPS, and metaverse are showing a nearly 27% annual growth in IIoT sensor networks [8] and are a hotbed for emerging technologies [3], [9], [10]. This phenomenon is characterized by 3VS, i.e., volume, variety, and velocity of sensors which are growing rapidly in DSM [11]. Moreover, the 3VS yields

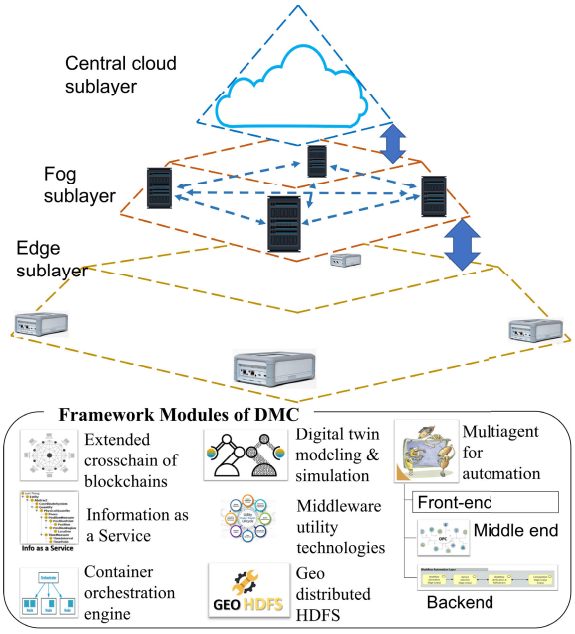


Fig. 1. DMC for Fukushima RTF, Japan.

excessive volume, variety, and velocity of data (3VD), big data [6], [11]. The significant growth in IIoT and the 3VD of 3VS poses a formidable challenge for acquiring and processing data while preserving the essential quality of service (QoS) requirements in DSM [3], [11], [12], [13], [14].

DMC platforms are one of the preferred cloud platforms for geodistributed networks [15], [16], [17], [18], including DSM. The University of Aizu's robot research group is working on the DMC shown in Fig. 1 for the Fukushima robot test field (RTF), which is the world's first and only known public all-in-one RTF facility to date. The RTF was established by the Japanese government to explore new opportunities for contributing to industrial evolution [19], [20], [21]. Universities, research institutions, and enterprises, including DSM, are the end-users of the DMC. However, DMC platforms are severely constrained by an IIoT sensor framework for DSM that guarantees indispensable QoS demands involving the 3VD of 3VS [3], [11], [12], [13], [14], [15].

The construction of DSM uses hierarchical and federated building blocks, as shown in Fig. 2. A DSM *factory network* comprises a number of *geodistributed factories*. Each *factory* comprises a number of *production lines* of *cells*, where a *cell* will typically involve an *array of IIoT sensors*. DSM involves various types of QoS-oriented demands, such as integrity, robustness, resource consumption, cost, and latency [3], [11], [12], [13], [14]. Among these, we focus on integrity and robustness in DSM. This means that DSM needs to affirm the integrity of its operation against the risk of breaching the security and privacy of valuable information (in the process of communications) and identities (confidentiality of components). With respect to robustness, scalability and interoperability should be preserved without compromising integrity. We can then note that security, privacy, scalability, and interoperability are essential multiobjective QoS requirements for maintaining the integrity and robustness of components and operations [17], [18], [19], [20], [21], [22].

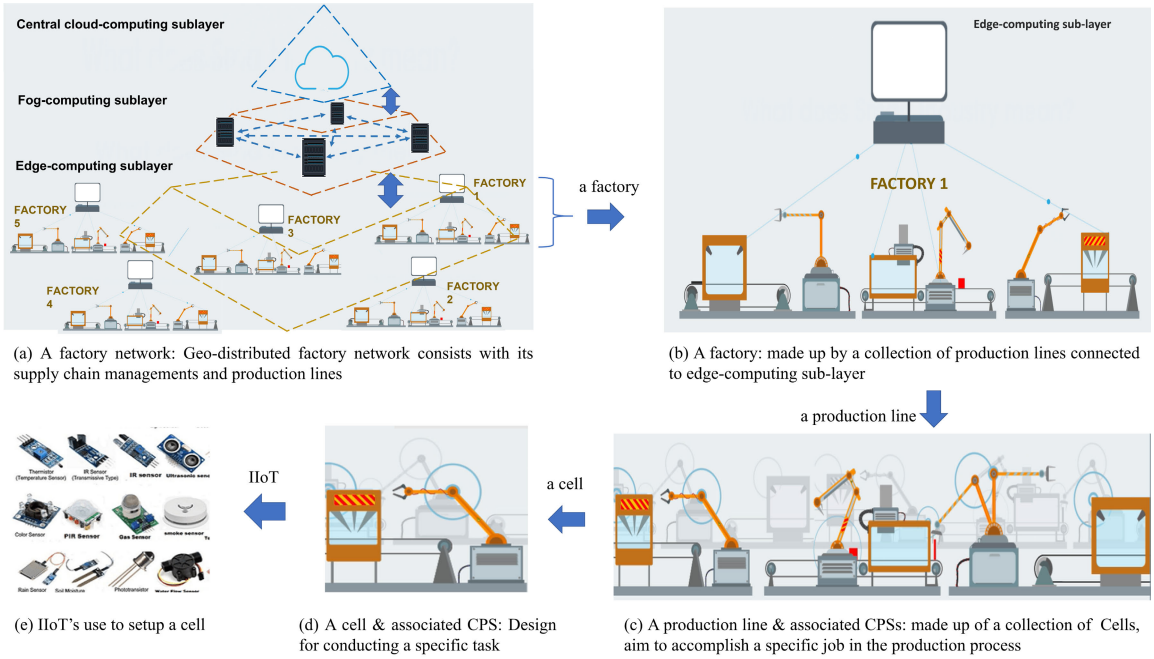
Blockchain is one of the most significant techniques available for addressing integrity concerns in the IoT domain, and blockchain-based solutions for smart industry are abundant [23], [24], [25], [26], [27], [28], [29], [30], [31], [32], [33], [34], [35], [36], [37], [38], [39], [40], [41], [42], [43], [44], [45], [46], [47], [48], [49], [50], [51], [52]. According to our survey of blockchain-based DSM solutions, 80% of the solutions are based on the on-chain approach and its variants for their key functional requirements, such as network commissioning and transactions [23], [24], [25], [32], [33], [34], [35], [36], [37], [38], [39], [40], [41], [42], [43], [44], [45], [46], [47], [48], [49], [50], [51], [52]. However, the first and second generations of blockchain solutions had severe constraints on scalability and interoperability and least provision for hierarchical distribution [53], [54], [55], [56]. Therefore, preserving scalability and interoperability in a hierarchically federated architecture without compromising security and privacy are critical concerns when using conventional blockchain-based solutions.

The third generation of blockchains, a.k.a. crosschains, has attracted both industry and academia. Moreover, crosschain technology complements the Web3 and metaverse paradigms, which aim to provide the foundation for a distributed ecosystem of decentralized applications [54], [55], [56], [57]. In addition, a crosschain ecosystem provides the necessary infrastructure for extending to a *federated distribution* of blockchain networks while simultaneously preserving essential QoS demands, such as security, privacy, scalability, and interoperability [54], [55], [57]. Furthermore, a hierarchically federated IoC facilitates the *hierarchical distribution* of QoS-aware federated crosschain networks [19], [21], [55], [57]. Therefore, an IoC is a preferred solution for a QoS-aware communication network for *hierarchically federated* DSM.

Moreover, concerning the holistic standpoint, ad hoc architectures for DSM have become a widespread practice. For instance, more than 85% of the selected blockchain-based literature covers ad hoc SAs without adequate AMP, model, or reasoning [23], [24], [25], [26], [29], [30], [31], [32], [33], [34], [35], [36], [37], [38], [39], [40], [41], [42], [43], [44], [45], [46], [47], [48], [49], [50]. Ad hoc solutions exhibit severe constraints on customizability and adaptability with respect to the viewpoint of software engineering and system architects. Nevertheless, we have learned that ad hoc solutions dedicated to either specific use cases (unique) or domains (homogeneous) are widespread. However, DSM involves heterogeneous manufacturing. Therefore, a top-down software AMP is a cognitive solution to addressing ad hoc issues. The SRA is a well-known holistic AMP [6], [15], [19], [58]. Therefore, we propose a novel crosschain-based SRA leading to an AMP for DSM called the *IoC of IIoFC* for DSM.

The SRA for the DSM is the industrial extension of the extended crosschain of blockchains (ECBs) [19]. The ECB provides two key representations: for the general public [21] and for industry. Our three key contributions are as follows.

- proposing an AMP for SRA modeling for DSM,
- reasoning an SRA from the proposed SRA model,
- deducing an SA from SRA and proposing pseudocode.



(a) A factory network: Geo-distributed factory network consists with its supply chain managements and production lines

(b) A factory: made up by a collection of production lines connected to edge-computing sub-layer

(c) A production line & associated CPSs: made up of a collection of Cells, aim to accomplish a specific job in the production process

(d) A cell & associated CPS: Design for conducting a specific task

(e) IIoT's use to setup a cell

Fig. 2. High-level view of distributed manufacturing network, including FAS, CPS, and SCM. (a) Factory network: geodistributed factory network consists of its SCMs and production lines. (b) Factory: made up by a collection of production lines connected to edge-computing sublayer. (c) Production line and associated CPSs: made up of a collection of cells, aiming to accomplish a specific job in the production process. (d) Cell and associated CPS: design for conducting a specific task. (e) IIoT's use to setup a cell.

To the best of our knowledge, this study is the first to propose an AMP for SRA modeling for a QoS-aware IIoT sensor network in DSM, that is, modeling for the *IoC of IIoFC*, SRA, SA, and a prototype for abstract representation. The proposed SRA and SA are abstract capability architectures.

Nomenclatue section outlines frequently used technical acronyms. The remainder of this article is organized as follows. Section II presents a literature survey. In Section III, we discuss the preliminaries and a motivation scenario. Section IV presents AMP and deduces the SRA, which forms the first part of the case study. Section V delves deeper into the case study, reasoning about the SRA model, and the derived SA, including pseudocode details and a discussion. Section VI presents the evaluation and its results. Finally, Section VII concludes.

II. Related Works

This section provides a *subjective literature survey* of blockchain-based manufacturing factory solutions relevant to our research objectives. With respect to the popularity of blockchain-based manufacturing factory solutions, we considered blockchain-based patents [23], [24], [25], [26], [27], [28] and literature [29], [30], [31], [32], [33], [34], [35], [36], [37], [38], [39], [40], [41], [42], [43], [44], [45], [46], [47], [48], [49], [50], [51], [52].[1] Our survey matrix was *"studying the blockchain-based manufacturing factory solutions for their top-down software architectural initiatives that facilitated hierarchically federated geodistribution while possessing QoS demands beyond default security and privacy."*

[1]The yearly lists of publications were: 2016 [28], 2017 [27], [35], [42], [51], 2018 [23], [26], [34], [36], 2019 [25], [33], [37], [38], [39], 2020 [24], [32], 2021 [29], [40], [48], 2022 [30], [31], [41], [44], [46], [49], [52], and 2023 [43], [45], [47], [50].

TABLE I
Classified Literature Works With Respect to Their Provision's for AMP, QoS, and Distributed Manufacturing

	Archi. Model	Process Model		QoS awareness		Distribution	
		SRA	SA	≤ 2	≥ 3	Local	Geo
$SG_{1.1}$ [23]–[25]	-	-	-	✓	-	-	✓
$SG_{2.1}$ [29]–[33]	-	-	✓	-	✓	[33]	[29]–[32]
$SG_{2.2}$ [34], [35]	-	-	✓	✓	-	-	✓
$SG_{2.3}$ [26], [36]–[50]	-	-	✓	✓	-	✓	-
$SG_{3.1}$ [51]	-	✓	-	✓	-	✓	-
$SG_{3.2}$ [27], [28]	✓	-	✓	✓	-	-	✓
$SG_{3.3}$ [52], our-work	✓	✓	✓	[52]	our-work	[52]	our-work

We observed the following taxonomy, which helps maintain the objectivity of this study. Our study investigated works under three main classes: *AMP for the solution* under software engineering, *number properties* under QoS awareness, and *ability to geodistribute* under hierarchically federated DSM. We then observed the following subgroups under the main classes. The subgroups of *AMP* were *modeling, SRA,* and *SA*, while the subgroups of *QoS awareness* were default security and privacy (*two*) or beyond default (*three/more than three*). Finally, *local* and *geo* were subgroups of *geodistribution* with/without hierarchical federation.

First, we analyzed the selected works and then sorted them with respect to their awareness of *AMP, QoS,* and *distribution*. Following that, we compiled them as presented in Table I, identifying three primary groups (G_1, G_2, and G_3 are based

on *AMP*) and seven subgroups (SGs are based on *QoS* and *distribution*). Our investigation is based on the taxonomy depicted in Table I. Therefore, we have assigned reference codes to subgroups to enhance the clarity. G_1 did not make minimal provision for AMP [23], [24], [25]. G_2 had ad hoc SAs without reasoning [29], [30], [31], [32], [33], [34], [35], [36], [37], [38], [39], [40], [41], [42], [43], [44], [45], [46], [47], [48], [49], [50]. Finally, G_3 possessed a minimal AMP with reasoning [27], [28], [51], [52] and our work.

G_1 and G_2 works do not have adequate AMP accounting in 26 out of 31 cases. However, G_2 refers to 23 works and is the largest group, whereas G_1 contains only $SG_{1.1}$, which refers to three patents and supports geodistribution [23], [24], [25]. All three of these studies adopted the on-chain technique, which performs crucial DSM activities inside the blockchain, such as commissioning the network and performing transactions. This means that all the member nodes participate in the decision-making process. In addition, on-chain does not have a provision for hierarchical distribution. Therefore, even though on-chain works are proposed for geodistribution, they make very limited provisions for scalability and interoperability.

The G_2 group contains three SGs: $SG_{2.1}$, $SG_{2.2}$, and $SG_{2.3}$. $SG_{2.1}$ comprises [29], [30], [31], [32], [33]. A unique observation of this subgroup is that it is one of only two subgroups making QoS beyond the default, the other being $SG_{3.3}$. One of these works [33] is limited to local factory deployment, with the remainder [29], [30], [31], [32] being proposed for geodistribution. Rožman et al. [29] is the only study that considers sidechain functionality with limited federated node distribution. However, the sidechain has constraints on the privacy of subunits of the hierarchically distributed node subsets. "Sharding" is a technique for enabling horizontal scaling and involves dividing the blockchain into smaller partitions and was used in [30]. In this context, [29], [30] are the only non-on-chain works in $SG_{2.1}$. The remaining studies are dominated by the on-chain-based methods.

$SG_{2.2}$ makes provision for geodistribution, whereas $SG_{2.3}$ is limited to local adoption. Subgroup $SG_{2.3}$ is the largest subgroup, comprising more than 50% (16 out of 31) works. This phenomenon implies that most researchers have preferred to use ad hoc SAs with respect to the holistic viewpoint and were limited to the default QoS provision. In [38] and [45], both on-chain and off-chain techniques were used. In an off-chain technique, certain functionalities of the on-chain architecture, such as batch processing of large data and payment channels, are offloaded. However, the constraints of on-chain traits, such as constraints in commissioning, operations, and scalability, were retained.

Particular advantages of works in G_1 and G_2, in contrast to those in G_3, are that they are easily understood and make proposals that are ready to be implemented because they involve fewer software design parameters in their architectures. However, their key disadvantages are that they have limited adaptability beyond the immediate DSM issue and are limited to unique or homogeneous use cases.

Group G_3 contains three subgroups that differ with respect to their modeling aspects: $SG_{3.1}$ contains only [51], which involves SRA, a generic architectural approach, makes provision only for the default QoS, and is limited to local adoption.

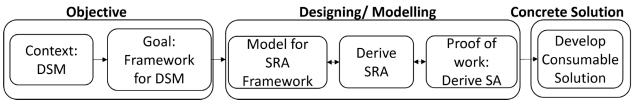


Fig. 3. Flow of architecture decision process for *Context* DSM.

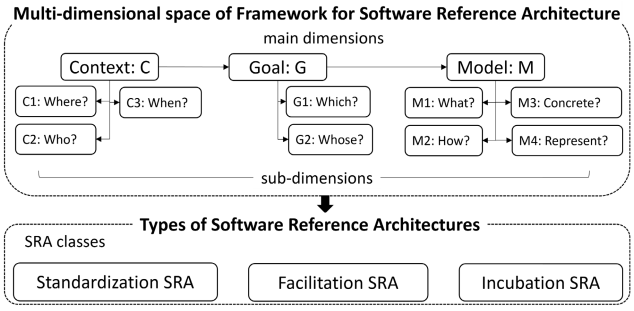


Fig. 4. Multidimensional space and SRA classes.

Subgroup $SG_{3.2}$ contains modeling for the solution and SA [27], [28]. These works facilitate the default QoS and can be adapted for geodistribution. The sharding technique is used by [28].

Finally, $SG_{3.3}$ incorporates [52] and our study. In [52], on-chain and off-chain techniques were used, enabling their architecture to offload certain computational duties from the main chain. However, constraints on the on-chain technique remain; only the default QoS provision is made, and only local adoption is available. Our proposed model, the *IoC of IIoFC*, uses an AMP, an SRA, and an SA, makes provision for four QoS factors, and can be adapted for heterogeneous DSM. The *IoC of IIoFC* is the only work that is custom-tailored for DSM's hierarchically federated distribution.

In summary, 80% of the works investigated used the on-chain technique [23], [24], [25], [32], [33], [34], [35], [36], [37], [38], [39], [40], [41], [42], [43], [44], [45], [46], [47], [48], [49], [50], [51], [52]. Their objectives include improving beyond default QoS requirements, such as privacy and security, and supporting DSM operations. However, as described above, on-chain approaches impose significant constraints on scalability, interoperability, and hierarchically federated distribution, ultimately severely affecting efficiency. Moreover, the unique disadvantages of G_3 studies are that they are complex and not ready for rapid deployment when compared with works belonging to G_1 and G_2. However, G_3 works mostly belong to the software framework category and can cover heterogeneous use cases when compared with ad hoc architectures. The specific benefits of the proposed *IoC of IIoFC* approach include the following: using an AMP that allows for adaptation and extension to accommodate diverse use cases, providing safeguards for integrity, including privacy and security, while maintaining the robustness, i.e., scalability and interoperability of modules and their submodules within a DSM network, and embracing a hierarchically federated architecture, enabling the delivery of tailor-made solutions to diverse DSM organizations.

III. Preliminaries and a Motivation Scenario

This section discusses the preliminaries and the motivation scenario for the DSM. Section III-A introduces the DMC. Section III-B presents the motivation scenario.

A. Decentralized Multicloud

DMC is the model-driven cloud platform proposed for the Fukushima RTF. A DMC is followed by an AMP, which comprises comprehensive modeling for the SRA, and SA [19]. First, we extended the work of Angelov et al. [58] on an AMP for the system-of-systems' problem requirement, which involves heterogeneous stakeholders. In this way, SRA modeling was performed to comply with the unified architectural framework solution.

Fig. 1 depicts a nonnormative view of the DMC, which comprises three cloud layers (i.e., edge, fog, and central) and nine key framework modules, and they are: ECB, container orchestration engine (COE), digital twin modeling and simulation (DTMS), middleware for adapting utility technologies (MUT), geodistributed Hadoop distributed file system (GD-HDFS), information as a service based on domain ontology and web services (ISOW), and multiagent for automation (MAA), with three submodules: *front end, middle*, and *back end*.

The edge, fog, and central cloud represent the DMC ecosystem. The ECB aims to guarantee the integrity and robustness of the DMC via a bipartite representation for the general public and for industry. General public representation was proposed in [21], and we present the proposal in the industry use case. ISOW facilitates domain ontology-based [59] web service composition [15] for knowledge management across the board. The COE is used for managing the scalability of end-users' custom environments (container orchestration) across the geodistributed DMC environment. The DTMS component is responsible for facilitating the digital twin requirements of the DMC. The MUT is used for adapting the required utility middleware technologies, including robotic operating systems and applications. The GD-HDFS component maintains a fault-tolerant geodistributed Hadoop distributed file system facility. The MAA manages the automation capability across the platform [60], [61], [62], [63]. The MAA front end handles all the requests from the end-users, the middle manages tasks automatically across the various components to handle their request inputs and outputs, and the back end is used for automating the required analytical and data science requirements (big data analytics, including deep learning, machine learning, and data mining) [64], [65].

B. Motivation Scenario

Company-A provides services for DSM clients with respect to FAS design, assembly, and maintenance, including SCM and CPS. A client of *Company-A* is *Company-P*, an electric vehicle manufacturing company that has a *geodistributed factory network*. Fig. 2 depicts the use case of SCM, FAS, and CPS of *Company-P*. Fig. 2(a) illustrates the SCM of the geodistributed five-factory network of *Company-P*, which uses three-layered DMC cloud infrastructure units: edge, fog, and central. Fig. 2(b) represents *Factory 1*, one of the five *Company-P* factories, which operates via an FAS and includes multiple *production lines*. The *production lines* communicate primarily with the DMC edge-computing layer at the factory level. Moreover, Fig. 2(c) represents one of the *production lines* and its associated CPS units of the many *production lines*

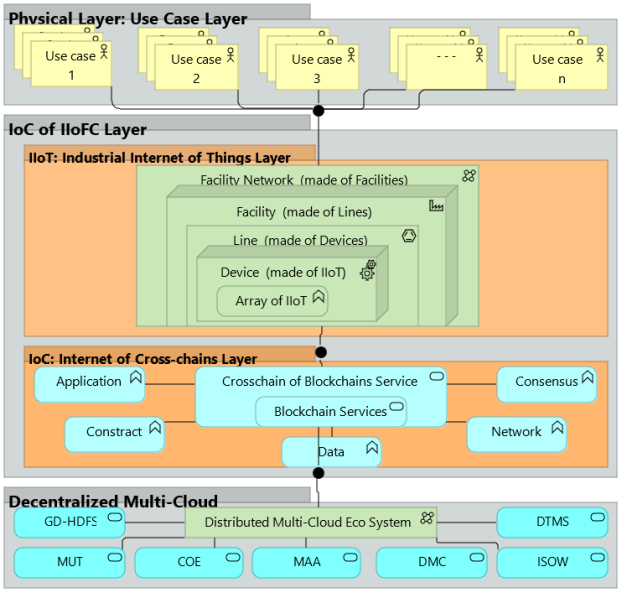


Fig. 5. Deduced normative view of M_4^{DSM}: SRA for DSM.

in *Factory 1*. *Production lines* contain multiple *cells* comprising CPS units and robotics accessories. Fig. 2(d) shows one of the *cells* and its associated CPS units in the *production line* shown in Fig. 2(c). A *cell* will usually include an *array of IIoT sensors*. Fig. 2(e) shows examples of *IIoTs*, which are the key building blocks in Fig. 2(d), which shows a *cell*.

Company-A must address two key concerns of *Company-P*. First, the integrity of their operation should be addressed because of the risk of breaching their communication network and the highly confidential identities of components (*factory network, factory, production line, cell*, and *IIoT*) of the DSM facility. Therefore, they should guarantee the security of communication and the privacy of respective components from a low-level *IIoT* to the entire *geodistributed network* of *Company-P*. Second, *Company-P* requires ensuring the robustness caused by a lack of confidence in network performance without compromising integrity. That is, *Company-P* should be able to scale up or down their component sets freely and interoperability across the range of components without compromising their integrity. Therefore, in this case, the DSM service provider *Company-A* reached out to the University of Aizu's robotic research group to address these issues and affirm the integrity (i.e., security and privacy) and robustness (i.e., scalability and interoperability) of *Company-A's* operations.

IV. AMP and Modeling for SRA for DSM

In this section, we discuss the AMP concept in Section IV-A, propose an AMP in Section IV-B, and deduce an abstract SRA for DSM in Section IV-C.

A. Background to the AMP for SRA Concept

This study's objectives are to propose an AMP for an SRA, deduce an SRA, derive an SA, and produce a proof-of-work prototype for developing concrete solutions for heterogeneous DSM facilities. Inspired by Angelov et al. [58], we used an SRA modeling technique for the next-generation system-of-systems' model [19].

Fig. 3 displays the three key stages of the AMP for modeling an SRA. They are *specifying objectives, designing/modeling*, and a *concrete solution*. The problem domain, called *Context*, is the *DSM* being considered. The *goal* is to *propose a framework for DSM*. The *designing/modeling stage* comprises three substages; *modeling for the SRA framework, deducing an SRA based on the model*, and *deriving a proof-of-work SA* based on the SRA. In this study, we developed a *prototype as a proof-of-work* based on the SRA and SA.

Fig. 4 shows the multidimensional space and three types of SRA, as inspired by [19] and [58]. The three key dimensions in modeling for SRA are *context (C), goal (G)*, and *model (M)*. Here, C involves C_1–C_3 subdimensions, G comprises G_1 and G_2, and M involves M_1–M_4 subdimensions. Next, we determined the type of SRA based on the respective multidimensional space values for C, G, and M. The particular AMP developed follows Fig. 4.

B. AMP for DSM

First, the *requirement* for the DSM is defined as follows.

Definition 1 (Requirement R_i and Constraint $c_{i,j}$): R_i denotes the ith use case of DSM, which requires addressing $c_{i,j}$ constraints and includes QoS demands set off against the *goals* of the DSM *objective*.

Here, DSM is the *context* of the problem domain, and R_i is one of the heterogeneous use cases of DSM. The $c_{i,j}$ constraints are applicable for all the heterogeneous use cases of DSM. c_1 specifies that the *proposed solution should be adaptable for the communication network requirements of DSM use cases, which means guaranteeing c_2 and c_3. c_2 refers to integrity, namely, the security and privacy of the operation*, and c_3 *refers to the robustness of the business flow, namely, scalability and interoperability.* Next, we define the *stakeholder*, where Fig. 4 depicts three main dimensions and their subdimensions.

Definition 2 (S_i: Stakeholder): S_i denotes the individual, group (individuals), organization, or group of organizations who are willing to use (adopt or adapt) the SRA model to satisfy the given R_i of DSM under the c_i constraints.

Definition 3 (C: Context): The SRA satisfies the R_i requirement of DSM S_i under constraints C_j and j for the problem definition. C involves the C_1–C_3 subdimensions.

Definition 4: C_1: Where Will it be Used?): C_1 refers to the DSM S_i of the SRA. Here, C_1 applies to an independent individual, many individuals (such as an organization/group of organizations belonging to a homogeneous DSM), or organizations belonging to the heterogeneous DSM.

Definition 5 (C_2: Who Defines it?): C_2 is focused on the modeling aspects of the SRA. Here, S_i of DSM drafts requirement specifications for the model, which includes end-to-end preparation for adapting the model by *groups of researchers, software companies*, and *policymakers*.

Definition 6 (C_3: When Is it Defined?): C_3 represents the SRA timing factor and includes three factors: *preliminary, classical*, and *hybrid*. *Preliminary* represents the components involved in software, such as techniques, technologies, and algorithms that are candidates for adoption, *classical* implies the components exist and are verified for use. The *hybrid* components exhibit both *preliminary* and *classical* characteristics.

Equation (1) represents C's main dimensions and subdimensions

$$\text{Context } C = \{C_1, C_2, C_3\} \tag{1}$$

$$C_1 = [\text{single, many, heterogeneous}] \tag{1a}$$

$$C_2 = [\text{req.-specifying-user, modeling-user}]$$
$$\text{req.-specifying-user} = C_1$$
$$\text{mod.-user} = (\text{research, software-comp, policymakers}) \tag{1b}$$

$$C_3 = [\text{preliminary, classical, hybrid}]. \tag{1c}$$

C, C_1, C_2, and C_3 adopted for DSM are C^{DSM}, C_1^{DSM}, C_2^{DSM}, and C_3^{DSM}. Equation (1), as adopted for DSM, results in (1'). In C_2, *Modeling-user* expands to *researcher, software company*, and *policymakers*

$$C^{\text{DSM}} = \left[C_1^{\text{DSM}}, C_2^{\text{DSM}}, C_3^{\text{DSM}}\right] \tag{1'}$$

$$C_1^{\text{DSM}} = [\text{heterogeneous}] \tag{1'a}$$

$$C_2^{\text{DSM}} = [\text{req.-specifying-user :: heterogeneous-org, modeling-user :: heterogeneous-org}] \tag{1'b}$$

$$C_3^{\text{DSM}} = [\text{classical}]. \tag{1'c}$$

According to (1'), S_i belongs to the S_i role described in C_1. In our case, it is *heterogeneous*. However, the SRA proposal should be adaptable. Therefore, the proposed C_3^{DSM} is *classical*.

Moreover, the *goal* G, its dimensions, and its subdimensions are specified via Definitions 7–9.

Definition 7 (G: Goal): The SRA objective is the *goal* G and may refer to single or multiple subgoals. Here, G refers to the requirement of S_i, defined by C_2, and involves two subdimensions G_1 and G_2.

Definition 8 (G_1: Which Stage Needs to be Defined?): Here, G_1 represents the class of the respective SRA. G_1 has three stages, namely, *standardization, facilitation*, and *incubation*. C_1 and C_3 contribute to G_1.

For instance, SRA is *standardization, facilitation*, or *incubation*, based on C_3 characteristics, which are defined by S_i, as one of *preliminary, classical*, or *hybrid*.

Definition 9 (G_2: Who Is Defined?): G_2 expresses the SRA ownership and purpose. G_2 is involved with C_1 and C_2.

G values are summarized by the following:

$$\text{Goal } G = \{G_1, G_2 : \text{subgoal}, \cup \text{ subgoal}\} \tag{2}$$

$$G_I = [C_1.C_3 : \text{standardization, facilitation, incubation}] \tag{2a}$$

$$G_2 = [\text{single}.C_2, \text{many}.C_2, \text{heterogeneous}.C_2]. \tag{2b}$$

For the DSM domain, G values become G^{DSM}, G_1^{DSM}, and G_2^{DSM}, with (2') being (2) adopted for DSM. Here, C_3^{DSM} is *hybrid*, i.e., G_1^{DSM} belongs to *facilitation* with *heterogeneous stakeholders*. The SRA goal to satisfy DSM S_i is R_i of *heterogeneous DSM*

$$G^{\text{DSM}} = \left\{G_1^{\text{DSM}}, G_2^{\text{DSM}} : \cup \text{ subgoal}\right\} \tag{2'}$$

$$G_1^{\text{DSM}} = [\text{Stakeholder:hetero::stage:facilitation}] \tag{2'a}$$

$$G_2^{\text{DSM}} = [\text{Stakeholder:hetero:OwnedBy:req-spec-user, Stakeholder:hetero::OwnedBy:model.-user}]. \tag{2'b}$$

Finally, M dimensions are defined as follows.

Definition 10 (M: Model): The abstract modeling dimension of SRA is defined as the specification for the design SRA. Here, M_1–M_4 are the subdimensions of M.

Definition 11 (M_1: What Is Described?): M_1 refers to the contents of the SRA. This involves custom-tailored information about the software, such as techniques, technologies, algorithms, connectors, and flow of information.

Definition 12 (M_2: How Many Layers?): M_2 refers to abstraction layers in the SRA and specifies three such layers: *aggregate*, *semidetail*, and *detail*. *Aggregate* has a single layer of component aggregation, *semidetail* has two to four aggregation layers, and *detailed* refers to five or more aggregation layers in the solution.

Definition 13 (M_3: How Concrete?): M_3 refers to the degree of abstraction in the SRA layers, which may be *abstract*, *semiconcrete*, or *concrete*. Information about the respective components is generic in an *abstract* description. At the *semiconcrete* level, the component description includes an abstract functional explanation, whereas *concrete* indicates a comprehensive description of the components.

Definition 14 (M_4: How to Represent?): M_4 discusses the SRA's possible levels of formalization and presentation of its semantics. M_4 offers *informal*, *semiformal*, and *formal* options. *Informal* is a general or incomplete graphical representation with the highest level of freedom to expand the art with ambiguity. Most components will be specified halfway through a *semiformal* presentation. However, modeling for the general solution will retain some freedom for ambiguity. *Formal* representation has less freedom for ambiguity compared with the other two presentations.

The M values are shown in the following:

$$\text{Model } M = \{M_1, M_2, M_3, M_4\} \tag{3}$$

$$M_1 = [\text{components, connectors, information flow}] \tag{3a}$$

$$M_2 = [\text{aggregated, semidetailed, detailed}] \tag{3b}$$

$$M_3 = [\text{abstract, semiconcrete, concrete}] \tag{3c}$$

$$M_4 = [\text{informal, semiformal, formal}]. \tag{3d}$$

M_1^{DSM}–M_4^{DSM} are the M values adapted for DSM. M_2^{DSM}–M_4^{DSM} values are *semidetailed*, *semiconcrete*, and *semiformal*, respectively. The (3) adopted for DSM is as follows:

$$M^{\text{DSM}} = \left\{M_1^{\text{DSM}}, M_2^{\text{DSM}}, M_3^{\text{DSM}}, M_4^{\text{DSM}}\right\} \tag{3'}$$

$$M_1^{\text{DSM}} = [\text{components, connectors, info. flow}] \tag{3'a}$$

$$M_2^{\text{DSM}} = [\text{semidetailed}] \tag{3'b}$$

$$M_3^{\text{DSM}} = [\text{semiconcrete}] \tag{3'c}$$

$$M_4^{\text{DSM}} = [\text{semiformal}]. \tag{3'd}$$

C. Case Study: Deduce an Abstraction Model and SRA for DSM

Our case study is discussed in this section and continued in Section V. First, we consider the modeling stage for deducing an abstract model for the DSM. Next, we deduce an SRA for the DSM.

We expand the variables of (3') and the identified layers of M_2^{DSM} as follows.

The *physical layer* contains the heterogeneous components of the DSM end-user: that is, the *IIoT sensors, cell, production line, factory*, and *factory network*. The *DMC ecosystem layer* dominates the collection of framework modules briefly discussed in Section III-A. The *IoC of IIoFC layer* is responsible for the operations dedicated to the ECB industrial use case, namely, the DSM business. The *IoC of IIoFC layer* involves two sublayers: the *IIoT* and *IoC* sublayers.

The *IIoT sublayer* refers to the physical appearance of the end-user hierarchical and federated IIoT use cases. This structure is such that the *facility network* comprises many *facilities*, each *facility* may comprise multiple *lines*, each *line* comprises a set of *devices*, and each *devices* consists of an *array of IIoT sensors*. The *IoC sublayer* represents the communication network, which is constructed from a crosschain and sets of blockchains. The *application, contract, consensus, network,* and *data* are appropriate properties associated with crosschains and blockchains.

Equation set (3") gives a representation of these values of the modeling specification adopted for the DSM domain. In (3") , we specified M_1^{DSM}–M_3^{DSM}. M_4^{DSM} can be represented by either of the two methods shown in Fig. 5 (normative representation of M_4^{DSM} (i.e., SRA) and Fig. 7 (nonnormative representation of M_4^{DSM} (i.e., high-level SA)

$$M_1^{\text{DSM}} = [\text{components, connectors\&flow}] \tag{3"a}$$

$$\begin{aligned} M_2^{\text{DSM}} = [&\text{Physical Layer: end-use cases,}\\ &\text{IoC of IIoFC Layer :}\\ &\quad\text{(IIoT sublayer, IoC sublayer)}\\ &\text{Decentralized Multi-Cloud Layer]} \end{aligned} \tag{3"b}$$

$$\begin{aligned} M_3^{\text{DSM}} = [&\\ &\text{Physical Layer: Heterogenous DSM use cases,}\\ &\text{IoC of IIoFC Layer:(}\\ &\quad\text{IIoT Sublayer:\{}\\ &\quad\quad\text{facility-nw[facility[line[device[IIoT]]]]}\\ &\quad\text{\}}\\ &\quad\text{IoC Sublayer:\{}\\ &\quad\quad\text{Cross-chain[app., cont., cons., } nw\text{, data],}\\ &\quad\quad\quad\text{[Blockchain (app., cont., cons.,}\\ &\quad\quad\quad\quad\text{nw, data)}\\ &\quad\quad\text{]}\\ &\quad\text{\}}\\ &\text{)}\\ &\text{DMC-Eco-Sys Layer: (DMC, GD-HDFS,}\\ &\quad\text{MUT, COE, ECB, MAA, DTMS, ISOW}\\ &\text{)}\\ &\text{].} \end{aligned} \tag{3"c}$$

We first consider the normative approach, where the ArchiMate modeling tool was used to design the SRA. The normative M_4^{DSM} comprises the three main layers described

in M_3^{DSM}, namely. the *physical, the IoC of IIoFC*, and the *DMC ecosystem* layers.

As shown in Fig. 4, *standardization, facilitation*, and *incubation* are three classes of abstract SRA models [19]. According to Definition 8, *facilitation SRA* refers to components that exist and are verified. In addition, (1') and (2') proposed a solution involving *classical* components and was aimed at heterogeneous DSM use cases. In (3') and (3"), layer descriptions are presented in *semidetailed, semiconcrete*, or *semiformal* forms.

Consequently, Fig. 5 shows the deduced SRA for the DSM. Fig. 5 contains the three main layers described in M_3^{DSM} of (3"c). The top layer dominates the use cases of DSM. The bottom layer represents the DMC. The middle layer, that is, the *IoC of IIoFC*, has two sublayers, namely, the *IIoT sublayer* and the *IoC sublayer*. The *IIoT sublayer* represents the physical infrastructure of the IIoT in various forms, from the basic to the geodistributed, and which is connected hierarchically in a federated manner. For example, the basic building block is an *array of IIoT sensors*. Above this, a *device* is made up of an *array of IIoT sensors*. Next, a *line* is made of multiple *devices*. A *facility* is made of multiple *production lines*. Finally, the *facility network* comprises many *facilities*, preparing for the communication network of facilities. Note that *IIoT sensor, device, line, facility*, and *facility network* are abstract terms used to give a generally applicable broad perspective of DSM. Fig. 6 shows the organizational view of the proposed DSM's *IoC of IIoFC* with respect to their physical and digital representation (DR).

V. Reasoning SRA Model: Derive SA and Pseudocode for Prototype (Continuing the Case Study)

This section continues the case study by reasoning about the SRA model, obtaining an SA in Section V-A, and presenting pseudocode for the prototype in Section V-B. Finally, Section V-C conducts a discussion.

A. Deduce SA for DSM

First, Section V-A1 describes the adoption of M_3^{DSM} and M_4^{DSM} for Scenario 1, as discussed in Section III-B. Next, Section V-A2 describes the derivation of an SA.

1) Adopt M_3^{DSM} and M_4^{DSM} for Scenario 1: For (3"c), the *IoC of IIoFC sublayer* of M_3^{DSM} comprises the *IIoT sublayer* and the *IoC sublayer*. As shown in (3"c), the IIoT and crosschain are the key building blocks for the proposed solution. Therefore, we adopt (3"c) to comply with Scenario 1.

According to Scenario 1 and presented in (3"c), *facility network, facility, line, device*, and *array of IIoT* sensors are the key elements of the *IIoT sublayer*. M_3^{DSM} is evaluated as *Company-P network, factory, production line, cell*, and *IIoT sensors*, resulting in the following:

$$\text{IIoT sublayer} = \{\text{Company-P}nw(\text{factory}(\text{prod-line}(\text{cell}(\text{IIoT}))))\}. \quad (3\text{"c.1})$$

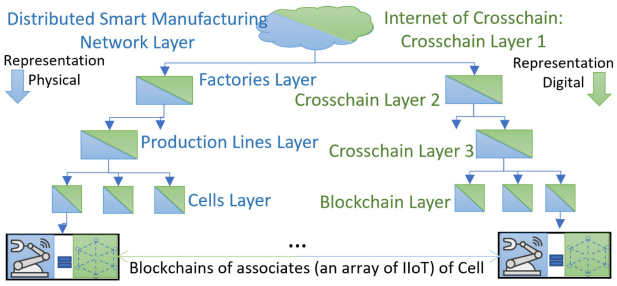


Fig. 6. Organization view of the proposed the *IoC of IIoFC* for DSM.

Moreover, the next key sublayer is the *IoC sublayer*, which contains the crosschain of (3"c). Equation (3"c.2) presents the adopted *IoC sublayer* for the scenario. Fig. 6 depicts the physical and digital presentations of the *Company-P network*

```
IoC sublayer = {
    Crosschain of Company-P nw, factory, prod-line
        (app., smart-contract, consensus,
                direct-decentralized, data),
        (Blockchain of Cell
            [app., smart-contract, proof of authority,
            indirect-centralized, data]
        )
}.                                              (3"c.2)
```

The crosschain plays an essential role in connecting many blockchains [54], [55], [57]. A frequently used technique for crosschains is the use of relay techniques that connect diverse blockchains and build an Internet network for the blockchain ecosystem, where Polkadot [57] and Cosmos [55] are two major crosschain providers. Polkadot networks are parachain and parathread blockchains, with Polkadot relay chain being used to connect parachains and parathreads securely [57]. Cosmos is powered by Tendermint consensus [55], which depends on the Tendermint core, connecting utility technologies and tools such as databases, necessary libraries, and webservers in the operation of blockchains. Therefore, the technologies behind Cosmos and Polkadot are candidates for the crosschain proposed in this solution.

Crosschain is used by the *Company-P network, factory*, and *production line* of the *IIoT sublayer*. The *cells* of the *IIoT sublayer* are networked using the blockchain technique, and the building blocks for a given *cell* are the *IIoT* sensors for that particular *cell*.

Candidate blockchain technologies include RapidChain [66], Lightning [67], and the Lite blockchain. Conventional blockchains exhibit severe delays in network commissioning and transaction. RapidChain, Lighting, and Lite blockchain technologies focus on transaction speed. RapidChain is a layer-1 on-chain and relies on sharding [66]. In contrast, Lightning is a layer-1 non-on-chain [67].

The properties of the cell blockchain described in (3'c) are substituted in accordance with (3"c.2). Fig. 6 shows an organization tree view representing the *cell layer* of the

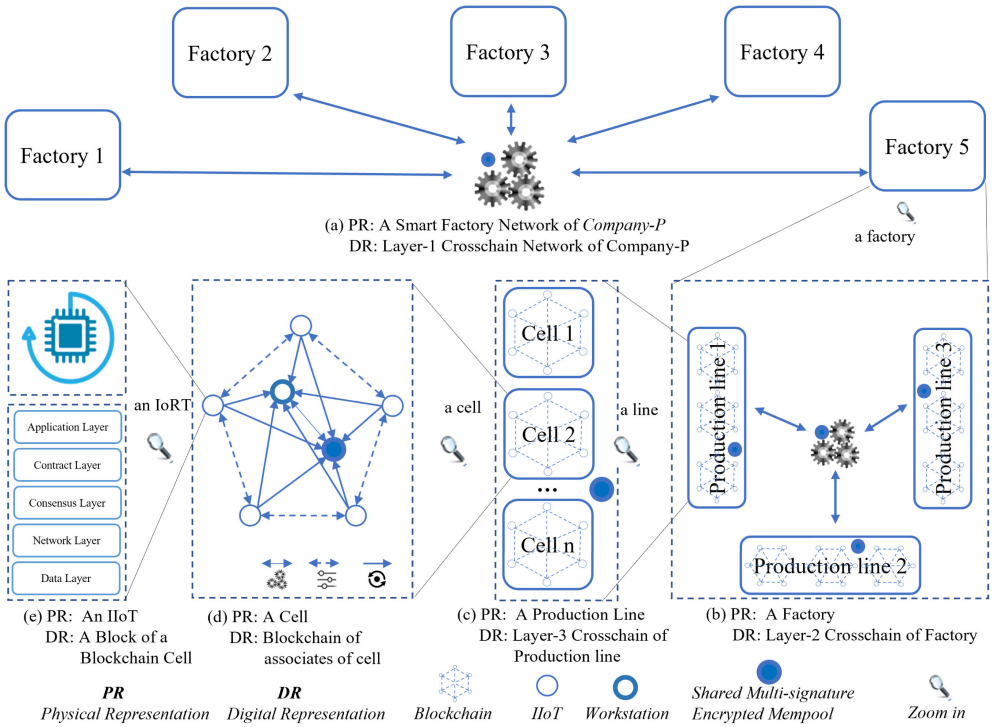


Fig. 7. Nonnormative view of M_4^{DSM}: abstract SA for DSM. (a) PR: a smart factory network of Company-P DR: layer-1 crosschain network of Company-P. (b) PR: a factory DR: layer-2 crosschain of factory. (c) PR: a production line DR: layer-3 crosschain of production line. (d) PR: a cell DR: blockchain of associates of cell. (e) PR: an IIoT DR: a block of a blockchain cell.

Company-P network. Each *blockchain cell* is dedicated to a predefined task associated with an *array of IIoT sensors*.

The overall system uses asymmetric encryption in communication [68] and modes reflecting the protocol of networked blockchains. That is, crosschain zones can be direct (decentralized), indirect (centralized), passive, or active modes [69]. In this way, the SA designer has the freedom to select adequate protocols for the custom requirement.

2) SA for M_4^{DSM} Scenario 1: As shown in Fig. 3, the third step of the *designing/modeling stage* is *proof-of-work: derive SA*. According to (3"c), M_3^{DSM} has three main layers, with the middle layer, the *IoC of IIoFC layer* having two sublayers. We adopt (3"c.1) and (3"c.2) as representative of these two sublayers, the *IIoT sublayer* and the *IoC sublayer*, to comply with the Scenario 1 use case presented in Section III-B.

The alternative M_4^{DSM}'s normative view, that is, developing an SRA for DSM, is shown in Fig. 5, As shown in Fig. 6, components of the *IIoT sublayer* are arranged hierarchically and are distributed in a federated manner. Therefore, we derived the DSM SA for Scenario 1 as depicted in Fig. 7, thereby maintaining freedom for flexibility in the desired pattern. Fig. 7(a)–(e) are mapped according to the components of (3"c.1), namely, *Company-P factory network, factory, production line, cell*, and *array of IIoT sensors*, respectively. Note that the subfigures (a)–(e) of Fig. 2 and Fig. 7, respectively, complement each other. The subfigures of Fig. 7 have views from the two perspectives, namely, physical representation (PR) and DR. PR refers to the physical appearance of the particular component, and DR means the digital terminology associated with the component.

The PR of Fig. 7(a) represents a geodistributed *Company-P factory network* dedicated to the SCM. As represented by DR, Figs. 6 and 7(a) are the layer-1 crosschain networks for five factories' crosschain networks. Each subcrosschain network or layer-2 crosschain with respect to Fig. 6 represents a geodistributed Company-P, namely, a DSM. We propose to connect these crosschains using a direct-decentralized method [69].

The PR of Fig. 7(b) represents a *factory* with three *production lines*. The DR of Fig. 7(b) is the layer-2 crosschain network shown in Fig. 6. As shown in Fig. 7(c), the PR is a factory's *production line*. The *production line* is made up of a number n of *cells*. Next, for DR, Fig. 7(c) is the layer-3 crosschain shown in Fig. 6 for the n blockchain *cells*. Here, the intercommunication of Fig. 7(b) with Fig. 7(c) depends on the factory situation. Therefore, we keep them open and informal at this stage.

Fig. 7(d) shows the PR of a *cell* for a given *production line*. The *cell* comprises five IIoT sensors, a workstation, and a multisignature shared memory pool. The DR of Fig. 7(d) represents the blockchain of associated *IIoT sensors* as per Fig. 6. The cell administrator determines the connection approach (such as *proof-of-authority, proof-of-stake, proof-of-work,* or *manual*), the IIoT sensors, and the workstation in the cell blockchain. The multisignature shared mempool (MSM) maintains all the records of data communication across the blockchain.

Fig. 7(e) shows the PR for an IIoT sensor and its associated microprocessor. With respect to DR, the figure shows one of the blocks of the given cell blockchain network.

B. Pseudocode for the Proposed Prototype

This section describes the generation of the *IoC of IIoFC* in terms of Algorithm 1, which comprises three procedures. The first procedure occupies Lines 1–4. The second procedure (Lines 5–17) invokes the *Industrial Internet of Federated Company* (*IIoFComp*) process. The third procedure (Lines 18–33) represents the class structure of the *IoC sublayer*, used by the second procedure.

First, the DSM administrator should declare the Company-P organization flow org_p, component constitutions cons_p, and requirements for preparing crosschains and blockchain variables within the Company-P network invocation $R_{C,B}$. The IoC of IIoFC results in a DSM communication network at the end of the procedure execution. The first procedure in Line 2 describes the abstract class of the *IoC sublayer*, which takes $R_{C,B}$ as input and results in a matching IoC_{NW} for that particular invocation. Next, at Line 3, the procedure invokes the *IIoT sublayer*, thereby preparing the IIoFC and resulting in DSM_{NW}. The first procedure ends in Line 4.

The second and third procedures elaborate the abstract definitions used in the first procedure. The second procedure describes the abstract crosschain class and its composition. Line numbers 6 and 7 represent the class variables, *data, contract, consensus, application,* and *network*. Here, *data, network, contract, consensus,* and *application* represent *transactions, binding contracts* (for a respectful chain), *a way of connecting blocks/nodes* (*proof-of-work, proof-of-stake, proof-of-authority,* or *manual*), and *an application use case for that chain*, respectively. The network variable involves a complex data structure, a subcrosschain, and subblockchains. The subcrosschain is used to connect individual factories or production lines. A subblockchain connects components of a cell. Line 16 returns IoC_{NW}, and the second procedure ends at Line 17.

Lines 18–33 describe the third procedure (*IIoT sublayer*), which prepares DSM_{NW} for Company-P. This procedure specifies the IIoFC preparation and uses org_p and cons_p to prepare the end result. Lines 19, 22, 24, and 26 indicate that the flow prepares a layer-1 crosschain, a layer-2 crosschain, a layer-3 crosschain, and a blockchain, respectively, as shown in Fig. 6. Company-P comprises p *factories*, q *production lines* in each *factory*, and r *cells* in each *production line*. Here, p, q, and r vary with respect to their component identities. Therefore, loops in Lines 21, 23, and 25 invoke *factory, production line*, and *cell*. For the cell (Line 2), the procedure adds k blocks, which are the IIoT sensors for that particular cell. In Line 28, the procedure creates an MSM for that cell, where the MSM holds all the records of transaction data within that blockchain cell. Line 32 returns the DSM_{NW} for Company-P, and Line 33 ends the third procedure.

Algorithm 1 IoC of IIoFC Procedure

Require: $Company\_P(Org\_p)$, $Components(cons\_p), CB\ req.(R_{C,B})$
Ensure: DSM_{NW} *of IoC of IIoFC*

```
1: procedure MAIN: IOC_OF_IIOFC(Com_p, cons_p)
2:     IoC_NW : IoC_SLayer → abs_Crosschain(R_C,B)
3:     DSM_NW : IIoT_SLayer ← prep_IIoFC(org_p, cons_p)
4: end procedure
5: procedure IOC_SLAYER: ABS_CROSSCHAIN(Comp_NW, R_c)
6:     data, contract, consensus, application
7:     procedure NETWORK: (i ‖ j ‖ k component)
8:         block of (company ‖ prod.line) component
9:         Crosschain: data, contract, consen, app, nw
10:        procedure BLOCKCHAIN: (k^th cell)
11:            block of cell components
12:            data, contract, consensus, app., nw
13:        end procedure
14:        return Network
15:    end procedure
16:    return IoC_NW
17: end procedure
18: procedure IIOT_SLAYER: PREP_IIOFC(org_p, cons_p)
19:    set up L1 crosschain network of Company_P
20:    Com_p: factory_nw, factory, prod_line, cell
21:    for i ← 1 to p # factories do
22:        set up L2 crosschain network of i^th factory
23:        for j ← 1 to q # of Prod.Lines do
24:            set up L3 crosschain nw of j^th prod.line
25:            for k ← 1 to r # of Cells do
26:                set up blockchain network of k^th cell
27:                add IIoT sensors to the k^th cell
28:                create shared MSM of k^th cell
29:            end for
30:        end for
31:    end for
32:    return DSM_NW
33: end procedure
```

C. Discussion

This section briefly discusses the achievement of objectives in the AMP and various aspects of QoS in the proposed method, together with the limitations of the method.

1) AMP, Model, SRA, and SA: The *designing/modeling stages* shown in Fig. 3 were conducted for a QoS-aware IIoT sensor network for a hierarchical and federated DSM facility. First, we proposed an AMP and a model for SRA in DSM in Section IV-B. Next, Section IV-C deduced an SRA for DSM from the proposed model. Section IV-C adopted values from Scenario 1 for the DSM variables in the SRA. Section V-A derived a *proof-of-work* SA for Scenario 1. Finally, Section V-B proposed pseudocode for the proposed prototype. This process achieved the aims of the software architectural *designing/modeling stage* shown in Fig. 3. This implies that the proposed AMP for DSM's QoS-aware IIoT sensor network was completed successfully.

2) PrivacycAwareness: The identities of end-users and devices across all the levels (*array of IIoT sensors, cell, production line, factory*, and *factory network*) are considered private and involve active privacy management. The *array of IIoT sensors* in a particular *cell* connects to a blockchain, and the given IIoT becomes one of the nodes of that particular blockchain. Asymmetric encryption has been proposed, which uses intracommunication and intercommunication in and out of the blockchain. Therefore, each IIoT sensor and end-user can access relevant network devices while ensuring end-to-end encryption for their communication.

In addition, the decentralized distributed ledger and consensus further strengthen privacy (ownership, validity, and authenticity) across the networks. Variables described in the AMP will be flexible to enable customizing as per the end-user's requirements. Therefore, the properties of blockchains and crosschains also have to be flexible to enable

customization. This inclusivity feature ensures end-user satisfaction without compromising their privacy.

3) Security Awareness: Security and privacy are two preferred aspects of blockchain-based applications. Here, security and privacy complement each other. Security of intercommunication and intracommunication of blockchains and crosschains is guaranteed via multiple methods at every level (*IIoT sensor, cell, production line, factory*, and *factory network*) of the network.

A consensus method, such as proof-of-authority, controls authorizing nodes in the blockchain network by the administration/management of the given factory's *production line*. The integrity of blockchains is strengthened by the decentralized ledgers, which are dedicated to the respective individual subunits in the DSM hierarchy. Generally, scalability and vulnerability are equally relevant. However, when blockchain networks are scaled up, the vulnerability risk is reduced.

4) Cross-Communication Awareness: As discussed above, the proposed crosschain is effective for seamless QoS-aware inter- and intracommunication. Therefore, the IIoT sensors of a particular cell of a given *production line* can communicate in and out of their smart factory network without compromising security or privacy.

The MSM is created primarily for recording transactions involving a specific group of nodes. For instance, suppose a given cell includes an MSM ledger. The MSM of that cell is shared within the nodes of that cell. In addition, the MSM may include transactions involving any outside node that is identified as an authenticated node. That particular MSM is then responsible for maintaining communication records of all the authenticated nodes. Moreover, every message is signed by the sender using *keyed hashing for the message-authenticate function* and encrypted by a *SHA256 password-based key-derivation function*.

Therefore, the MSM facilitates maintaining a ledger that possesses integrity: namely, the security and privacy of inter- and intracommunication of nodes. Moreover, according to Sections VI, MSM is a proven cost-effective, energy-efficient, and traffic-efficient scalable ledger.

5) Scalability Awareness: Scalability is severely limited in conventional blockchains. However, the provisions for hierarchically federated distribution facilitate scaling up the network without impacting the overall network performance. Moreover, MSM adoption improves transaction throughput and cost considerably. Therefore, a given DSM can scale up or down its components (such as *factories* in a *factory network, production lines* in a *factory*, *cells* in a *production line*, and *IIoT sensors* in a *cell*) and their subnetworks without compromising the QoS of the overall network.

6) Limitations: The proposed AMP and SRA are flexible and can be adapted or extended for heterogeneous DSM use cases within the scope of the parameters in the modeling process. However, the system architect can decide the crosschain and blockchain types. In addition, the limitations of those technologies may be reflected in the proposed method. Moreover, QoS awareness tends to vary with respect to technologies selected for the parameters proposed in the model.

VI. Evaluation

We conducted experiments to investigate the efficiency of the proposed *IoC of IIoFC*, when compared with an existing preferred blockchain-based DSM technique. In our literature review in Section II, we observed that 24 out of 30 methods used on-chain as the key technique for smart factory use cases [23], [24], [25], [32], [33], [34], [35], [36], [37], [38], [39], [40], [41], [42], [43], [44], [45], [46], [47], [48], [49], [50], [51], [52]. We therefore used an on-chain-based DSM to compare with our proposed method.

A. Preparation

In this section, we discuss the evaluation matrix we adopted and the experimental setup.

1) Evaluation Matrix: We prepared an evaluation matrix based on the key fundamental tasks involved in the DSM.

A DSM network involves two main element types: that is, nodes and transactions. These two elements engage in two network task types: module commissioning (building the network, a one-time process) and transaction processing (i.e., communication, which is repetitive).

Three quantitative features of these two task types are the *computation cost of commissioning modules, the processing time for the transaction procedure*, and *data involved in traffic congestion*. Therefore, we conducted empirical experiments to study (i.e., examine and analyze) the relative efficiency of the proposed method against the on-chain method using the following evaluation matrix to examine the efficiency of the proposed method for six experiments.

1) Section VI-B describes Experiments 1 and 2, which investigate the *efficiency of computation cost in commissioning modules*, first in the single-cell mode and then in the multiple-cell mode, in the DSM infrastructure.
2) Section VI-C describes Experiments 3 and 4, which investigate the *efficiency of the processing time for the transaction procedure*, first in the single-cell mode and then in the multiple-cell mode.
3) Section VI-D describes Experiments 5 and 6, which investigate the *efficiency of data involving traffic congestion*, first in the single-cell mode and then in the multiple-cell mode.

2) Experiment Setup: We prepared the proposed *IoC of IIoFC* and on-chain networks to simulate the DSM.

We prepared both the proposed *IoC of IIoFC* and the on-chain network as DSM simulations. To maintain objectivity, we used the same *number of nodes* (cells, production lines, company, and company network), the same *complexity* for the proof-of-work technique in preparing blocks of nodes, and the same *local IP addresses* to prepare network. We simulated the components of the distributed DSM environment as nodes of crosschains and blockchains that simulated the *company network, company, production line, cell,* and *IIoT sensors* as parallel-processed web applications with different local IP addresses for each component. The experiments were conducted using a Postman Version 10.14.8, Flask Version 2.2.3, Windows 11 Pro 64, Intel Core i7-12700H 2.3-GHz processor, and 32-GB RAM environment.

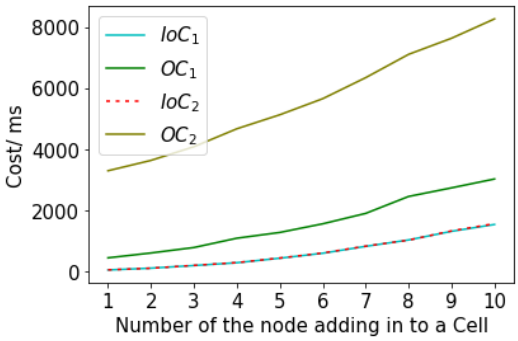


Fig. 8. Computation cost of commissioning node and cell modules.

B. Efficiency of Computation Cost in Commissioning Modules

We conducted two experiments to evaluate the computation cost for the commissioning modules. The *cell* is the basic building block in DSM. *IIoT sensors* are the nodes of a cell. First, we set up a *cell network* for a *production line* belonging to a *company* in a *company network*. In Experiment 1, we simulated the commissioning of a *cell* with up to *ten IIoT sensor nodes*, a *production line* incorporating such *cells*, a *company* using that *production line*, and a *company network*. We gradually increased the number of nodes per *cell* and determined the computation cost of making a new block of a given node type.

In Experiment 2, we prepared another *cell* in that *production line* and subsequently increased the number of nodes and determined the computation cost of adding a new block of the given node type. Fig. 8 gives the results of both the experiments.

The results of Experiment 1 are given by IoC_1 (our system) and OC_1, and the results of Experiment 2 are given by IoC_2 and OC_2. Note that IoC_1 and OC_1 have gradually increased computation costs when the number of nodes in the network increases. However, the IoC_1 curve remains lower than the OC_1 curve throughout. These two observations imply that IoC_1 retains the lowest computation cost for commissioning a cell in a DSM network. This phenomenon is caused by the proposed IoC_1 block addition being a streamlined process through its pooling technique. The node that introduces a novel block broadcasts its new block information to the pool members of that cell. This technique avoids the threads of individual nodes repeatedly checking on changes in the network. However, the node members of OC_1 are required to connect, add a new block (i.e., mine), and check for updates in the cell network as discrete tasks. This will increase the computation cost of commissioning a cell in an on-chain network.

Experiment 2 considered adding a new cell parallel to the existing cell. The results of Experiment 2 are given as IoC_2 and OC_2. Here, IoC_2 overlapped with IoC_1, while OC_2 maintained a significantly large gap when compared with IoC_1, IoC_2, and OC_1. The overlapping of IoC_1 and IoC_2 is caused by the architecture; that is, a given cell is an independent module in the DSM network. Moreover, adding (mining) and updating the respective ledger is an independent task dedicated to that cell. Therefore, only members of that cell are affected, and only records of members of that particular cell are maintained. Therefore, IoC_1 and IoC_2 have the same computation cost when preparing a cell. However, OC_2 retains a considerably higher cost compared with the other three tests throughout, and the cost of OC_2's first member is close to the cost of the last member for OC_1. Next, OC_2 shows gradually increasing computation cost when adding new blocks. This phenomenon is caused by the architecture of the on-chain method, where every node in the network is connected to the prior adjacent node, and every node is required to check for updates for themselves as discrete tasks. Moreover, according to the on-chain architecture, members of adjacent cells are dependent on each other for checking updates. This phenomenon resulted in a considerable cost increase when adding a new cell parallel to the existing cell in a DSM.

Therefore, according to the results in Fig. 8 of Experiment 1 and Experiment 2, the proposed method maintains the lowest computation cost of commissioning modules.

Therefore, according to the results in Fig. 8 for Experiments 1 and 2, the proposed method can be expected always to have a lower computation cost for commissioning modules than the baseline method.

C. Efficiency of the Processing Time for the Transaction Procedure

We conducted our third and fourth experiments to investigate the processing time for transaction procedures. Experiment 3 involves transaction procedures in a one-cell environment, and Experiment 4 involves transactions in a multiple-cell environment. For Experiment 3, we set up a simulated cell with ten IIoT sensors and performed transactions between the members of the cell with up to ten nodes. Each transaction comprised the following six types of record: *"from," "to," "data," "timestamp," "message,"* and *"signature."* To maintain objectivity, we limited the transaction *"data"* to a fixed string of five characters. *"From"* indicated who sent the message and *"to"* the receiver of the message. Messages were encrypted via SHA256 and signed by the sender. We began with two members and gradually increased the number of nodes involved in performing transaction procedures and recording the data associated with the transactions. Experiment 4 was conducted with an additional cell being added to the previously added cell. Transactions were subsequently performed between the members of the second cell, and the transaction data were recorded.

Fig. 9 presents the results for Experiments 3 and 4, where IoC_3 and OC_3 represent the results of Experiment 3 and IoC_4 and OC_4 represent the results of Experiment 4. For Experiment 3, IoC_3 maintains a line parallel to the x-axis, whereas OC_3 shows a linear increase for an increasing number of nodes. This shows that the number of nodes does not affect the cost of the transaction processes for the proposed method (i.e., the transaction process cost remains constant, irrespective of the number of nodes per cell.) This is one of the benefits of using an MSM. Nodes in a given cell are initialized with an MSM acknowledged by the members of the cell, and its address is shared by all the member nodes. Therefore, the transaction procedures maintain a nearly constant processing time and are independent of the number of nodes. In contrast,

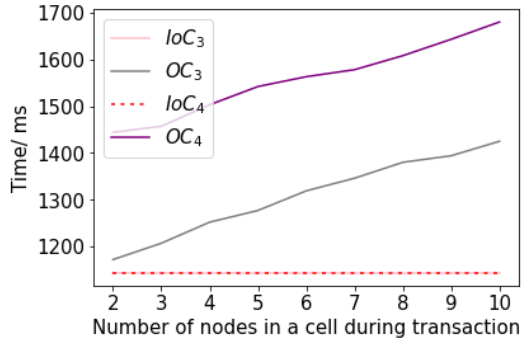


Fig. 9. Processing time of transaction procedure.

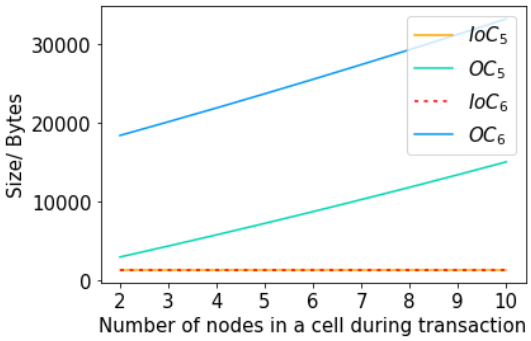


Fig. 10. Trafficked data during transaction.

each node of the on-chain must maintain an individual MSM ledger and remain up to date with the records of transactions for all the nodes of the network. Therefore, the cost of the transactions will increase linearly with respect to the number of nodes in an on-chain network.

The results of Experiment 4 are given by IoC_4 and OC_4 in Fig. 9. Here, IoC_4 overlaps with IoC_3, whereas the OC_4 curve remains above the other three curves throughout. Experiment 2 was conducted in terms of a second cell. Therefore, the network already contains nodes from a previous cell, and we can note that IoC_4 remains constant and overlaps with IoC_3. This phenomenon can be attributed to the fact that according to the proposed architecture of *IoC of IIoFC*, the second cell maintains its own MSM. Therefore, the transaction processing time is independent of the number of cells and nodes. However, OC_4 shows a large gap throughout compared with the other three curves, with its initial value being close to the last value of OC_3. Because the on-chain network requires the maintenance of individual MSMs, with each MSM having to remain up to date with all transactions in all the nodes, it will have a linearly increasing processing cost for the second cells.

The results of Experiments 3 and 4 shown in Fig. 9 demonstrate that the proposed *IoC of IIoFC* transaction procedure can be expected to have a lower computation cost than an on-chain transaction procedure, independently of the number of nodes and cells.

D. Efficiency of Data Involving Traffic Congestion

Experiments 5 and 6 investigated traffic congestion when performing transactions. We assumed that traffic-congested data would be proportional to the exchanged data volume when invoking the relevant services in a transaction procedure.

TABLE II
AVERAGE RATES OF INCREMENT TEST VALUES DEPICTED IN FIGS. 8–10

	Commissioning Modules (ms/node)		Transaction Process (ms/node)		Data in Traffic (bytes/node)	
	IoC	OC	IoC	OC	IoC	OC
One cell	166	287	0	31	0	1508
Two cells	169	553	0	31	0	1848

Therefore, we collected information about the data during Experiments 3 and 4 (discussed in Section VI-C). Experiment 5 collected data communicated between a one-cell environment, and Experiment 6 recorded information when the DSM involved the second cell in a two-cell environment.

Fig. 10 shows the results for Experiments 5 and 6, where IoC_5 and OC_5 represent the results of Experiment 5 and IoC_6 and OC_6 represent the results of Experiment 6.

In Fig. 10, IoC_5 is constantly low, and OC_5 linearly increases with respect to the x-axis. This implies that the data involved in traffic congestion for the proposed *IoC of IIoFC* method are low and independent of the number of nodes in a cell. This can be attributed to the MSM in this method, where read and write transaction information is recorded in a ledger shared by cell members. However, the nodes in an on-chain method must maintain individual MSMs dedicated to that node and record all the transactions of all the cell members. Therefore, the data contribution to traffic congestion will linearly increase in proportion to the number of nodes in a cell.

Note also that IoC_6 overlaps with IoC_5, with the OC_6 curve maintaining a considerable gap compared with the other three curves. The overlapping effect is caused by the cell modules of the proposed architecture being independent of each other. Therefore, the architecture does not have to maintain the internal records of other cells. Therefore, the data contribution to traffic congestion will remain independent of the number of nodes and cells. However, the on-chain architecture maintains records for all the transaction data across all the nodes of the network. Therefore, the data contribution to traffic congestion in the on-chain method is proportional to the number of nodes and the number of cells.

Consequently, Fig. 10 demonstrates that the data population for traffic within the proposed *IoC of IIoFC* method is independent of the number of nodes and cells and is always expected to be lower than that for the on-chain technique.

Table II summarizes the average rates of increment for test values from experiments conducted in Sections VI-B–VI-D. Therefore, according to the results of the experiments discussed in Sections VI-B–VI-D, the proposed *IoC of IIoC* method efficiency outperformed the baseline on-chain method with respect to the *computation cost of commissioning modules and their members, transaction processing time,* and *traffic congestion*. In particular, the *transaction processing time* and *traffic congestion* were independent of the number of nodes and cells.

VII. CONCLUSION

The objective of *IoC of IIoFC* is to alleviate the constraints of the current blockchain-based DSM. Therefore, we observed

that it requires a model-driven SRA that preserves essential QoS demands, such as the integrity and robustness of the IIoT framework for DSM. In pursuit of this objective, our article systematically unfolds the key components of our approach. In Section IV-B, we introduced the AMP tailored specifically for DSM. Subsequently, in Section IV-C, we deduced an SRA framework from the foundations laid by the AMP model. Section V-A derived an SA from the SRA, using Scenario-1 as a case study. To demonstrate the practical application of our work, Section V-A presented a real-world Scenario-1 case study, wherein we derived an SA from the SRA. Finally, Section V-B presented the pseudocode for the *IoC of IIoFC*. These steps are summarized in Fig. 3 and together confirm the theoretical and practical existence of a solution for DSM. Moreover, empirical experiments have validated the efficiency of our proposed *IoC of IIoFC* framework, demonstrating its ability to outperform conventional on-chain methods in key DSM tasks. Therefore, we conclude that the proposed *IoC of IIoFC* framework provides infrastructure for integrity and robustness, namely, seamless security, privacy, scalability, and interoperability.

As we look ahead to the future, our ongoing research focuses on extending our Web 3.0-based third-generation blockchain solution. Specifically, we are developing *IoC of IIoFC as a Service* and exploring its applications within the emerging industrial metaverse.

REFERENCES

[1] *IoT Market Size Worldwide 2017–2025 | Statista*. Accessed: Nov. 25, 2022. [Online]. Available: https://www.statista.com/statistics/976313/global-iot-market-size/

[2] *Regional Industrial Internet of Things Market Size 2017–2025*. Accessed: Nov. 25, 2022. [Online]. Available: https://www.statista.com/statistics/1102164/global-industrial-internet-of-things-market-size/

[3] B. Alotaibi, "Utilizing blockchain to overcome cyber security concerns in the Internet of Things: A review," *IEEE Sensors J.*, vol. 19, no. 23, pp. 10953–10971, Dec. 2019.

[4] H. Darvishi, D. Ciuonzo, E. R. Eide, and P. S. Rossi, "Sensor-fault detection, isolation and accommodation for digital twins via modular data-driven architecture," *IEEE Sensors J.*, vol. 21, no. 4, pp. 4827–4838, Feb. 2021.

[5] M. A. Ramírez-Moreno et al., "Sensors for sustainable smart cities: A review," *Appl. Sci.*, vol. 11, no. 17, p. 8198, 2021. [Online]. Available: https://www.mdpi.com/2076-3417/11/17/8198

[6] A. James, A. Seth, and S. C. Mukhopadhyay, "IoT system design-a project based approach," in *IoT System Design*. Berlin, Germany: Springer, 2022, pp. 9–33.

[7] H. Darvishi, D. Ciuonzo, and P. S. Rossi, "A machine-learning architecture for sensor fault detection, isolation, and accommodation in digital twins," *IEEE Sensors J.*, vol. 23, no. 3, pp. 2522–2538, Feb. 2023.

[8] S. Vitturi, C. Zunino, and T. Sauter, "Industrial communication systems and their future challenges: Next-generation Ethernet, IIoT, and 5G," *Proc. IEEE*, vol. 107, no. 6, pp. 944–961, Jun. 2019.

[9] S. Carrara, "The birth of a new field: Memristive sensors. A review," *IEEE Sensors J.*, vol. 21, no. 11, pp. 12370–12378, Jun. 2021.

[10] A. James, A. Seth, and S. C. Mukhopadhyay, "LoRa communication based IoT system," in *IoT System Design*. Berlin, Germany: Springer, 2022, pp. 167–191.

[11] M. S. Akbar, Z. Hussain, Q. Z. Sheng, and S. Mukhopadhyay, "6G survey on challenges, requirements, applications, key enabling technologies, use cases, AI integration issues and security aspects," 2022, *arXiv:2206.00868*.

[12] A. Samanta and T. G. Nguyen, "Quality-driven energy-efficient big data aggregation in WBANs," *IEEE Sensors Lett.*, vol. 6, no. 8, pp. 1–4, Aug. 2022.

[13] A. Samanta and J. Tang, "Dyme: Dynamic microservice scheduling in edge computing enabled IoT," *IEEE Internet Things J.*, vol. 7, no. 7, pp. 6164–6174, Jul. 2020.

[14] A. Samanta, T. G. Nguyen, T. Ha, and S. Mumtaz, "Distributed resource distribution and offloading for resource-agnostic microservices in industrial IoT," *IEEE Trans. Veh. Technol.*, vol. 72, no. 1, pp. 1184–1195, Jan. 2023.

[15] T. Hannadige, "Architecture for intelligent big data analysis based on automatic service composition," Ph.D. dissertation, Dept. Comput. Sci. Inf. Syst., Univ. Aizu, Aizuwakamatsu, Japan, 2019.

[16] S. C. Mukhopadhyay, N. K. Suryadevara, and A. Nag, "Wearable sensors for healthcare: Fabrication to application," *Sensors*, vol. 22, no. 14, p. 5137, 2022. [Online]. Available: https://www.mdpi.com/1424-8220/22/14/5137

[17] A. James, A. Seth, and S. C. Mukhopadhyay, "Cloud computing for IoT systems," in *IoT System Design*. Berlin, Germany: Springer, 2022, pp. 193–203.

[18] M. Fu et al., "Environmental intelligent perception in the industrial Internet of Things: A case study analysis of a multicrane visual sorting system," *IEEE Sensors J.*, vol. 23, no. 19, pp. 22731–22741, Oct. 2023.

[19] A. Siriweera and K. Naruse, "Survey on cloud robotics architecture and model-driven reference architecture for decentralized multicloud heterogeneous-robotics platform," *IEEE Access*, vol. 9, pp. 40521–40539, 2021.

[20] *Policies Cabinet Office*. Accessed: Sep. 19, 2023. [Online]. Available: https://www8.cao.go.jp/cstp/english/society5_0/index.html

[21] A. Siriweera and K. Naruse, "Internet of cross-chains: Model-driven cross-chain as a service platform for the Internet of Everything in smart city," *IEEE Consum. Electron. Mag.*, vol. 12, no. 3, pp. 85–97, May 2023.

[22] T. H. Akila, I. Paik, and S. Siriweera, "QoS-aware rule-based traffic-efficient multiobjective service selection in big data space," *IEEE Access*, vol. 6, pp. 48797–48814, 2018.

[23] K. Wolfson, A. Natanzon, and J. Shemer, "Validation of sensor data using a blockchain," U.S. Patent 10 778 426, Sep. 15, 2020.

[24] B. Moeller, "Management of a reliable industrial control system via dedicated cellular network," U.S. Patent 17 003 157, Jul. 15, 2021.

[25] C. H. Cella, G. W. Duffy, J. P. McGuckin, and M. Desai, "Methods and systems for data collection of machine signals utilizing a distributed ledger for analytics and maintenance using the industrial Internet of Things," U.S. Patent 16 684 757, Jun. 4, 2020.

[26] U. B. Posts, "Industrial data detection block chain network architecture based on edge computing and detection method," CN Patent 8 380 652, Feb. 19, 2019.

[27] N. M. Smith et al., "Blockchains for securing IoT devices," U.S. Patent 11 290 324, Mar. 29, 2022.

[28] C. A. Kohlhepp, "Methods for an autonomous robotic manufacturing network," U.S. Patent 10 152 760, Dec. 11, 2018.

[29] N. Rožman, J. Diaci, and M. Corn, "Scalable framework for blockchain-based shared manufacturing," *Robot. Comput.-Integr. Manuf.*, vol. 71, Oct. 2021, Art. no. 102139.

[30] N. Gao, R. Huo, S. Wang, T. Huang, and Y. Liu, "Sharding-hashgraph: A high-performance blockchain-based framework for industrial Internet of Things with hashgraph mechanism," *IEEE Internet Things J.*, vol. 9, no. 18, pp. 17070–17079, Sep. 2022.

[31] Y. Li, Y. Chen, K. Zhu, C. Bai, and J. Zhang, "An effective federated learning verification strategy and its applications for fault diagnosis in industrial IoT systems," *IEEE Internet Things J.*, vol. 9, no. 18, pp. 16835–16849, Sep. 2022.

[32] C. Zhang, G. Zhou, H. Li, and Y. Cao, "Manufacturing blockchain of things for the configuration of a data- and knowledge-driven digital twin manufacturing cell," *IEEE Internet Things J.*, vol. 7, no. 12, pp. 11884–11894, Dec. 2020.

[33] K. Zhang, Y. Zhu, S. Maharjan, and Y. Zhang, "Edge intelligence and blockchain empowered 5G beyond for the industrial Internet of Things," *IEEE Netw.*, vol. 33, no. 5, pp. 12–19, Sep. 2019.

[34] A. Vatankhah Barenji, Z. Li, and W. M. Wang, "Blockchain cloud manufacturing: Shop floor and machine level," in *Proc. Eur. Conf. Smart Objects, Syst. Technol.*, Jun. 2018, pp. 1–6.

[35] Z. Jadidi, A. Dorri, R. Jurdak, and C. Fidge, "Securing manufacturing using blockchain," in *Proc. IEEE 19th Int. Conf. Trust, Secur. Privacy Comput. Commun. (TrustCom)*, Dec. 2020, pp. 1920–1925.

[36] A. Kapitonov, I. Berman, V. Bulatov, S. Lonshakov, and A. Krupenkin, "Robonomics based on blockchain as a principle of creating smart factories," in *Proc. 5th Int. Conf. Internet Things, Syst., Manag. Secur.*, Oct. 2018, pp. 78–85.

[37] J. Wan, J. Li, M. Imran, and D. Li, "A blockchain-based solution for enhancing security and privacy in smart factory," *IEEE Trans. Ind. Informat.*, vol. 15, no. 6, pp. 3652–3660, Jun. 2019.

[38] L. Bai, M. Hu, M. Liu, and J. Wang, "BPIIoT: A light-weighted blockchain-based platform for industrial IoT," *IEEE Access*, vol. 7, pp. 58381–58393, 2019.

[39] A. Vatankhah Barenji, Z. Li, W. M. Wang, G. Q. Huang, and D. A. Guerra-Zubiaga, "Blockchain-based ubiquitous manufacturing: A secure and reliable cyber-physical system," *Int. J. Prod. Res.*, vol. 58, no. 7, pp. 2200–2221, Apr. 2020.

[40] M. Sandborn, C. Olea, S. Hays, and J. White, "An architecture for component authentication using secure cyber-physical information and blockchain," in *Proc. 6th Int. Conf. Fog Mobile Edge Comput. (FMEC)*, Dec. 2021, pp. 1–7.

[41] G. Manogaran, M. Alazab, P. M. Shakeel, and C.-H. Hsu, "Blockchain assisted secure data sharing model for Internet of Things based smart industries," *IEEE Trans. Rel.*, vol. 71, no. 1, pp. 348–358, Mar. 2022.

[42] N. Teslya and I. Ryabchikov, "Blockchain-based platform architecture for industrial IoT," in *Proc. 21st Conf. Open Innov. Assoc. (FRUCT)*, Nov. 2017, pp. 321–329.

[43] A. A. Khan et al., "Data security in healthcare industrial Internet of Things with blockchain," *IEEE Sensors J.*, early access, May 11, 2023, doi: 10.1109/JSEN.2023.3273851.

[44] Y. Jiang, Y. Zhong, and X. Ge, "IIoT data sharing based on blockchain: A multileader multifollower Stackelberg game approach," *IEEE Internet Things J.*, vol. 9, no. 6, pp. 4396–4410, Mar. 2022.

[45] C. Mazzocca, N. Romandini, M. Mendula, R. Montanari, and P. Bellavista, "TruFLaaS: Trustworthy federated learning as a service," *IEEE Internet Things J.*, early access, Jun. 5, 2023, doi: 10.1109/JIOT.2023.3282899.

[46] M. Shahjalal, M. M. Islam, M. M. Alam, and Y. M. Jang, "Implementation of a secure LoRaWAN system for industrial Internet of Things integrated with IPFS and blockchain," *IEEE Syst. J.*, vol. 16, no. 4, pp. 5455–5464, Dec. 2022.

[47] J. Cui, N. Liu, Q. Zhang, D. He, C. Gu, and H. Zhong, "Efficient and anonymous cross-domain authentication for IIoT based on blockchain," *IEEE Trans. Netw. Sci. Eng.*, vol. 10, no. 2, pp. 899–910, Mar. 2023.

[48] K. Yu, L. Tan, M. Aloqaily, H. Yang, and Y. Jararweh, "Blockchain-enhanced data sharing with traceable and direct revocation in IIoT," *IEEE Trans. Ind. Informat.*, vol. 17, no. 11, pp. 7669–7678, Nov. 2021.

[49] W. Wang, Y. Zhang, J. Gu, and J. Wang, "A proactive manufacturing resources assignment method based on production performance prediction for the smart factory," *IEEE Trans. Ind. Informat.*, vol. 18, no. 1, pp. 46–55, Jan. 2022.

[50] O. T. Sanchez et al., "An IIoT-based approach to the integrated management of machinery in the construction industry," *IEEE Access*, vol. 11, pp. 6331–6350, 2023.

[51] F. Zhang, M. Liu, and W. Shen, "Operation modes of smart factory for high-end equipment manufacturing in the internet and big data era," in *Proc. IEEE Int. Conf. Syst., Man, Cybern. (SMC)*, Oct. 2017, pp. 152–157.

[52] Y. Zhang, B. Li, J. Wu, B. Liu, R. Chen, and J. Chang, "Efficient and privacy-preserving blockchain-based multifactor device authentication protocol for cross-domain IIoT," *IEEE Internet Things J.*, vol. 9, no. 22, pp. 22501–22515, Nov. 2022.

[53] B. Huang et al., "BoR: Toward high-performance permissioned blockchain in RDMA-enabled network," *IEEE Trans. Services Comput.*, vol. 13, no. 2, pp. 301–313, Mar. 2020.

[54] *Research at Web3 Foundation—Research at W3F*. Accessed: Jan. 19, 2023. [Online]. Available: https://research.web3.foundation/en/latest/

[55] *Cosmos: The Internet of Blockchains*. Accessed: Jan. 19, 2023. [Online]. Available: https://cosmos.network/

[56] L. Cao, "Decentralized AI: Edge intelligence and smart blockchain, metaverse, web3, and DeSci," *IEEE Intell. Syst.*, vol. 37, no. 3, pp. 6–19, May 2022.

[57] *Polkadot: Web3 Interoperability | Decentralized Blockchain*. Accessed: Jan. 21, 2023. [Online]. Available: https://polkadot.network/

[58] S. Angelov, P. Grefen, and D. Greefhorst, "A framework for analysis and design of software reference architectures," *Inf. Softw. Technol.*, vol. 54, no. 4, pp. 417–431, Apr. 2012. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0950584911002333

[59] B. T. G. S. Kumara, I. Paik, J. Zhang, T. H. A. S. Siriweera, and K. R. C. Koswatte, "Ontology-based workflow generation for intelligent big data analytics," in *Proc. IEEE Int. Conf. Web Services*, Jun. 2015, pp. 495–502.

[60] T. H. Akila, S. Siriweera, I. Paik, and B. T. G. S. Kumara, "QoS-aware traffic-efficient web service selection over BigData space," in *Proc. IEEE Int. Conf. Comput. Inf. Technol. (CIT)*, Dec. 2016, pp. 197–203.

[61] T. H. A. S. Siriweera, I. Paik, and B. T. G. S. Kumara, "QoS and customizable transaction-aware selection for big data analytics on automatic service composition," in *Proc. IEEE Int. Conf. Services Comput. (SCC)*, Jun. 2017, pp. 116–123.

[62] I. Paik and T. H. A. S. Siriweera, "Automating big data analysis based on deep learning generation by automatic service composition," in *Proc. IEEE Int. Conf. Data Sci. Adv. Analytics (DSAA)*, Oct. 2019, pp. 610–611.

[63] T. H. A. S. Siriweera, I. Paik, B. T. G. S. Kumara, and K. R. C. Koswatta, "Intelligent big data analysis architecture based on automatic service composition," in *Proc. IEEE Int. Congr. Big Data*, Jun. 2015, pp. 276–280.

[64] A. Siriweera, I. Paik, and H. Huang, "Constraint-driven complexity-aware data science workflow for AutoBDA," *IEEE Trans. Big Data*, early access, Mar. 13, 2023, doi: 10.1109/TBDATA.2023.3256043.

[65] T. H. A. S. Siriweera, I. Paik, and B. T. G. S. Kumara, "Constraint-driven dynamic workflow for automation of big data analytics based on GraphPlan," in *Proc. IEEE Int. Conf. Web Services (ICWS)*, Jun. 2017, pp. 357–364.

[66] M. Zamani, M. Movahedi, and M. Raykova, "RapidChain: Scaling blockchain via full sharding," in *Proc. ACM SIGSAC Conf. Comput. Commun. Secur.*, Oct. 2018, pp. 931–948.

[67] J. Poon and T. Dryja, "The Bitcoin lightning network: Scalable off-chain instant payments," Tech. Rep., 2016.

[68] S. Nakamoto, "Bitcoin: A peer-to-peer electronic cash system," *Decentralized Bus. Rev.*, 2008.

[69] H. Jin, X. Dai, and J. Xiao, "Towards a novel architecture for enabling interoperability amongst multiple blockchains," in *Proc. IEEE 38th Int. Conf. Distrib. Comput. Syst. (ICDCS)*, Jul. 2018, pp. 1203–1211.

Akila Siriweera (Member, IEEE) received the B.Sc. degree from the University of Peradeniya, Peradeniya, Sri Lanka, in 2005, and the M.Sc. and Ph.D. degrees in computer science and engineering from the University of Aizu, Aizuwakamatsu, Japan, in 2016 and 2019, respectively.

His current research interests include Web 3.0, system-of-systems modeling, and automated big data analysis.

Dr. Siriweera is a TC Member of the IoT Group at the IEEE Consumer Technology Society. He has received several outstanding awards in his academic and industry careers.

Keitaro Naruse (Member, IEEE) is a Full Professor at the University of Aizu, Aizuwakamatsu, Japan. He has specialized in swarm robots and applications for agricultural-robotic systems and interface systems for robots in disaster responses. Nowadays, he works for design/model, DevOps, and standardized networked-distributed intelligent robot systems with heterogeneous sensors and robots. Various applications include service-robot systems, Factory Automation System (FAS), and intelligent disaster-response robot systems tested at the Robot Test Field (RTF).

Mr. Naruse's research team has received several awards during various international robot competitions.