IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 12 October 2022, accepted 25 October 2022, date of publication 28 October 2022, date of current version 16 November 2022.

Digital Object Identifier 10.1109/ACCESS.2022.3217794

SURVEY

Applications of Blockchain in Business Processes: A Comprehensive Review

WATTANA VIRIYASITAVAT[1], (Senior Member, IEEE), LI DA XU[2], (Fellow, IEEE), DUSIT NIYATO[3], (Fellow, IEEE), ZHUMING BI[4], (Senior Member, IEEE), AND DANUPOL HOONSOPON[1]

[1]Chulalongkorn Business School, Chulalongkorn University, Bangkok 10330, Thailand
[2]Department of Information Technology and Decision Sciences, Old Dominion University, Norfolk, VA 23529, USA
[3]School of Computer Science and Engineering, Nanyang Technological University, Singapore 639798
[4]Department of Civil and Mechanical Engineering, Purdue University Fort Wayne, Fort Wayne, IN 46805, USA

Corresponding author: Danupol Hoonsopon (danupol@cbs.chula.ac.th)

ABSTRACT Blockchain (BC), as an emerging technology, is revolutionizing *Business Process Management* (BPM) in multiple ways. The main adoption is to serve as a trusted infrastructure to guarantee the trust of collaborations among multiple partners in trustless environments. Especially, BC enables trust of information by using Distributed Ledger Technology (DLT). With the power of smart contracts, BC enforces the obligations of counterparties that transact in a business process (BP) by programming the contracts as transactions. This paper aims to study the state-of-the-art of BC technologies by (1) exploring its applications in BPM with the focus on how BC provides the trust of BPs in their lifecycles; (2) identifying the relations of BPM as the need and BC as the solution with the assessment towards BPM characteristics; (3) discussing the up-to-date progresses of critical BC in BPM; (4) identifying the challenges and research directions for future advancement in the domain. The main conclusions of our comprehensive review are (1) the study of adopting BC in BPM has attracted a great deal of attention that has been evidenced by a rapidly growing number of relevant articles. (2) The paradigms of BPM over Internet of Things (IoT) have been shifted from persistent to transient, from static to dynamic, and from centralized to decentralized, and new enabling technologies are highly demanded to fulfill some emerging functional requirements (FRs) at the stages of design, configuration, diagnosis, and evaluation of BPs in their lifecycles. (3) BC has been intensively studied and proven as a promising solution to assure the trustiness for both of business processes and their executions in decentralized BPM. (4) Most of the reported BC applications are at their primary stages, future research efforts are needed to meet the technical challenges involved in interoperation, determination of trusted entities, confirmation of time-sensitive execution, and support of irreversibility.

INDEX TERMS Blockchain, business process management, smart contracts, trust assurance.

I. INTRODUCTION

Business Process Management (BPM) has been prevalent since the late 1990s [86]; it is regarded as a confluence to bridge information technology and management science [3]. BPM is used to manage, compose, orchestrate, analyze, and diagnose business processes (BPs). Note that a BP is constructed by a set of activities or tasks that should be coordinated; while BPM becomes a key driver for the management of BPs [118]. With the intensified globalization and the proliferation of interoperations, BPM has been developed according to technological advances at all levels and domains of businesses in an unprecedented scale [75]. Nowadays, BPs are becoming service-oriented progressively and executions and communications of businesses are standardized for seamless interactions and integrations [158]. The rapid growth of Internet of Thing (IoT) technology has accelerated service-oriented businesses to utilize microservices; enterprises have

The associate editor coordinating the review of this manuscript and approving it for publication was Fu Lee Wang.

been highly pressured to improve their BPs by enabling the dynamic collaborations among business partners' services in highly turbulent business environments [12].

In open environments, sensitive data flows across distributed stakeholders, and this raises the concern in assuring security and privacy [6]. Conventional BPM tools centralized the controls of data flows and executions of BPs to assure overall security and privacy [92]. However, the centralized control was vulnerable to single point of failure and encountering the risk when the executer of a BP becomes malicious [131]. The lack of trust creates hesitation in BP participation and consequently inhibits the wide use of services. This lack is originated from that BP executions are relied on one entity [153]. Furthermore, centralized BPM shown some unsatisfactory performance such as latency and messaging delay [7], [122]. On the other hand, when BPs are distributed and decentralized, an adopted BPM tool must (1) *be flexible* to deal with changes, dynamics and uncertainties of business processes and stakeholders and (2) *be scalable* to managing and executing BPs successfully when the numbers of available online services are changing dynamically [166], [167], [168]. The challenges to BPM were briefed below with a comprehensive elaboration in section III and IV.

1) In centralized BPM, a sole entity has full control of BP executions; business partners are obligated to trust this entity for task executions and data protection [145].
2) In decentralized BPM, the ownership of a BP is shared. No entity acquires full authority of entire BP executions. Each partner is allowed to access and process information under assigned responsibilities [79]. Thus, it is necessary for BPM to monitor and validate executions during runtime [109].
3) With a rapid growth of IoT, many researchers argued that BC would be widely adopted to manage IoT-enabled business processes cost-effectively [169], [170], [171]. Therefore, BPs will involve greater BC-based services. Mechanisms to assure trust of such services among disparate partners becomes an essential element of modern BPM.
4) Proprietary and heterogeneity of services are the main causes of inconsistency and incompatibility, hindering interoperations in a large scale.
5) The implementations of BC-based services are heterogeneous implying different cost, performance, and delays to process and confirm BC transactions. Utilizing the services needs to be well justified to address these variants as well as uncertainties. For example, undesired consequences of time delays in a time-sensitive BP should be seriously addressed.

Assuring trust of information inside BP interoperations is one of the critical requirements in modern BPM; trust assurances are also required in the executions of tasks [145]. Since BPM tends to be decentralized, BC is naturally considered as the appropriate mechanism to establish trust repository among partners [65]. Additionally, BC-based applications can offer auxiliary functions to assist BPM. BC technology has brought numerous opportunities to advance BPMs. It was originated as a technology underlying cryptocurrencies but its applications have been drastically expanded to various domains. Research has been conducted progressively. Nowadays, BC is viewed as an promising solutions as trusted infrastructure, being capable of empowering conventional BPMs in the sense that service-oriented BPs can be managed, composed, planned, and coordinated in decentralized and distributed network with trust and with the full consideration of dynamics, uncertainties, scalability, disturbances, and heterogeneity [134]. Together with IoT, BC not only establishes trust to BPs but also expands the landscape of BPM greatly. The advance of BPM is pivoted by BC from managerial and strategical perspectives. It can be applied to solve the aforementioned challenges. For example, BC smart contracts can ensure correctness of BP executions. It can be configured to be a trusted repository and will be use as a trusted source for auditing. However, BC implementation inside BPM is not an easy task. Exclusive BC characteristics pose difficulties to be applied to BPMs. For instance, we need to deal with varying degree of trust and heterogeneity of each BC implementation, and uncertainties of time for transaction confirmation [155]. Research is progressively conducted to tackle these problems.

II. RELEVANT SURVEYS AND SLR OUTCOME

In this section, relevant surveys were summarized, their limitations on the coverage of BP and BCT were discussed, and our research methods were proposed to fill the gap in the understanding state of the art to advance BCT for BPM.

A. RELEVANT SURVEYS

In the past, a few surveys on BC technology have been published ranging from general objectives to analysis, assessment, and classification of diversified applications in different domains. Here, the conclusions from these surveys were justified to differentiate our work. The surveys unanimously agreed that BC has big potentials in many domains such as Supply Chain Managements (SCM), healthcare, transportation, manufacturing, smart cities, etc. Existing surveys mostly follow Systematic Literature Review (SLR) methods [31], [55], [111], [150]. They showed similarity in classifying BC applications, identifying emerging research areas, and understanding current and future development. However, most of them were confined their discussions relevant to specific needs or domains of interest. Within BPM domain, few surveys were found and the most relevant one was conducted by Mendling et al. [96]. It was found that BC could be utilized to improve BPM performance in each phase of lifecycle. The role of BC was discussed, and the challenges of implementation were identified at the aspects of networks, security assurances, and resource wastes specifically related to Proof-of-Work (PoW), a consensus algorithm used in Bitcoin. Later surveys were identified including Lauster et al. [78], Garcia-Garcia et al. [53] and Victor and Corentin [132]. The coverage of these surveys is limited and does not carry out exhaustive search, implying limited in contribution within specific, or interested, characteristics. Nonetheless, the survey by Medling et al. [96] is considered outdated. The number of the articles has been tripled since 2018.

B. RESEARCH METHODOLOGY

It is essential to provide the most updated survey in this field. This survey aims to provide a general and comprehensive content with a broader scope of recent and significant works to gain better understanding of the recent development in the field. To the authors' knowledge, this article would be the first survey for a crossover of BC and BPM in discussing the challenges, opportunities, and solutions. At the end, Table 3 was provided to understand relevant works by others, and this helped classify new works based on BCT development to its application in BPM. Moreover, the solutions to the following research questions were used as guides to examine the roles of BCT in BPM:

1) What aspects of BPs that can take advantages of BC?
2) Which challenges of BC associated with BPM characteristics and lifecycle have been reported in existing literature?
3) Which solutions have been proposed for the applications of BC in BPs and how they are implemented and evaluated? And
4) What are current and future research directions for the BC applications in BPM?

From the previous surveys in the past section, we notice that most recent works on BCT in BPM were of special interest, and SLR was adopted to classify and analyze the works based on their contributions to the targeted challenges and tasks. We adopt SLR approach [172], [173] in this study. The processes of SLR is shown in Fig. 1.

Search string: First, we identify and collect relevant works from multiple sources, the keywords in Table 1 were used for the search strings,

$$S_1 = (\vee_{i=1}^{8} A_i) \wedge (\vee_{i=1}^{7} B_i) \quad (1)$$

where S_1 was the search strings; A_i and B_iC_i were the keywords in two categories in Table 1; V referred to the sum operation.

TABLE 1. Searching keywords.

A	B
Blockchain	Business process
Distributed ledger	Workflow
Shared ledger	Business process management
Smart Contract	BPM
Cryptocurrency	Cross-organizational business
Bitcoin	Inter-organizational business
Ethereum	Distribute business process
Hyperledger	

Data sources: These included main academic databases. Keywords are used to search and collect the works on the applications of BCT in BPM from reputational academic databases listed here: *IEEE Xplore*, *ScienceDirect*, *Scopus*, *Web of Science, ACM Digital Library, SpringerLink, Wiley Inter-Science, Google Scholar*, and some other leading journals, proceedings, and workshops that had the themes of BCT and BPM. At this stage, 1,245 documents were identified when eq. (1) was applied to search these data sources by using the following fields: title, abstract, and keywords.

Inclusion and Exclusion Criteria: To determine the quality and relevancy of the documents, we developed criteria to evaluate the documents. They will be included if they at least have one of the criteria:

1) BCT or smart contract were adopted in BPM lifecycle.
2) BCT and smart contract were used as a framework for managing business processes.
3) BCT and smart contract were evaluated or analyzed as a solution to BPM.
4) BCT and smart contract were treated as a complete, or part of, solution in BPM.
5) BCT and smart contract were integrated to advance BPM.

To obtain most recently progress and focus on quality works, some none or less relevant, or low-quality articles were discarded. The following criteria were used for this exclusion.

1) BCT or BPM was just discussed in general with limited details.
2) It was published in a low or no ranking journals or proceedings such as some listed in Beall.
3) It has no reviewers.
4) Some were published before 2008 that are considered outdated. Note that some early but significant works were still included to understand the histories of BCT and BPM.

After the above criteria were applied, a total of 1,245 documents were reduced to 268 documents.

Quality Assessment: After that, a document's overall quality and relevance were evaluated further based on the following criteria:

1) Did it present a novel approach or conclude a new finding?
2) Was the presented works validated?

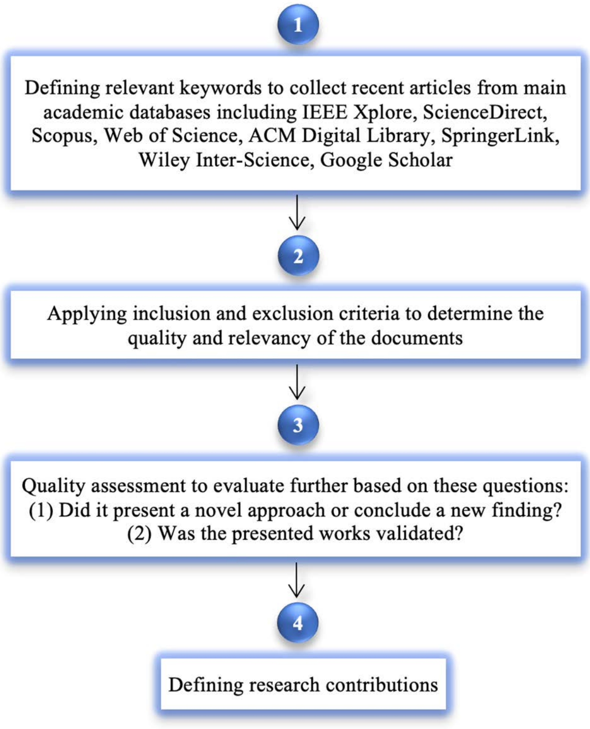
After applied, 64 articles are selected which will be classified into 10 categories summarized in Table 3.

Research Contributions: Most existing surveys address on specific areas and in-depth analysis towards BCT and BPM. Our survey seeks to make sense of the collection of the state-of-the-art BCT application in BPM, organizing based on SLR framework. Furthermore, key research directions in the context of BCT and BPM are identified and discussed. Researchers and practitioners may rely their works on this reference. The major contributions are as follows:

1) We conduct a comprehensive review on the application of BCT in BPM. Apart from reviewing the state-of-the-art research, we analyze and evaluate existing research in 10 perspectives along with the mentioned challenges, characteristics, and lifecycle.

2) By conducting a survey, we analyze from the empirical evidence for systematic assessment and comparison. The outcome makes researchers aware of current research attempts and enables them to realize the directions for future research.
3) This survey can be referred by not only BPM community regarding BPM potentials, but also BCT community the knowledge of BCT in BPM. This could encourage the wide range of research opportunities and practices.

Our attempt is to systematize important literature by identifying analysis benchmarks towards BC challenges, BPM characteristics, and lifecycle. This survey is organized as follows: Section III and IV discusses these benchmarks in detail. The collection of existing articles will be assessed and discussed towards the benchmarks. In section V, the survey is comprehensively conducted organized into ten categories. Section VI provides discussions and future research directions concluded from existing literature. Other important issues are also addressed. The survey is then concluded in section VII.


1 Defining relevant keywords to collect recent articles from main academic databases including IEEE Xplore, ScienceDirect, Scopus, Web of Science, ACM Digital Library, SpringerLink, Wiley Inter-Science, Google Scholar

2 Applying inclusion and exclusion criteria to determine the quality and relevancy of the documents

3 Quality assessment to evaluate further based on these questions: (1) Did it present a novel approach or conclude a new finding? (2) Was the presented works validated?

4 Defining research contributions

FIGURE 1. Steps in systematic literature review (SLR).

III. BC PRINCIPLES AND CHALLENGES IN BPM

BC is a relatively new technology and its implications to businesses are bewildering to many stakeholders [66]. Initially, BC was defined as a *distributed public ledger* [71]. It consisted of blocks and each block is a place to hold transactions recorded as data. A new block had a special header pointing to a precedent block that forms a chain. BC is constructed by combining relevant technologies [103]; for example, it exploits a cryptographic *hash* as a pointer so that any alteration in precedent blocks would have a cascade effect to subsequent blocks, which results in an inconsistency of pointer (hash) values. In such a way, frauds or unauthorized modifications would be detected. Immutability is preserved. When new transactions are introduced in a BC, they must be validated by nodes. Depending on BC types, nodes are recognized as miners or validators. The operation for validations is called *consensus,* which preserves integrity of shared ledgers [134]. *Consensus algorithm* can be tailored so that corresponding BC might tolerate faulty or malicious nodes at certain degree.

BC type is classified into private, permissioned, and public, which are defined based on ledger sharing schemes and the ways consensuses are performed. Table 2 provides a brief comparison of BC types [136]. Private BC has the highest degree of centralization. It is transparent when all BPs are within the sovereignty of a sole enterprise. Public BC is fully decentralized; it allowed any stakeholder to participate in all processes and maintain ledgers. It had the highest degree of openness with coming with the cost of assuring privacy and performance. Permissioned BC is balanced between private and public types by obtaining benefits of the two. Permissioned BC is not fully decentralized; it involves in more trust assumptions than public BC. Nodes called validators are chosen from pre-defined conditions to form a consortium. BC type must be chosen carefully and appropriately. Stakeholders must be well-aware for BPM to interact with various BCs [45].

TABLE 2. Comparison of public, private, and permission blockchains [136].

Type	Cost per Transaction	Performance		Trust	Scalability	Openness
		Throughput	Latency			
Public	High	Low	High	High	High	High
Private	Low	High	Low	Low	Low	Low
Permissioned	Fair	Fair	Fair	Fair	Fair	Low

BC intrinsic properties creates (1) trust and (2) provide natural resistance to single point of failure. When an attacker tries to modify past transactions, majority of nodes that hold the same view of shared ledgers will detect and nullify such actions. Note that BC integrates benefits from other technologies such as cryptography, database, and distributed controls to achieve its properties. BC disruptive potentials to BPM are rooted from its profound properties in constituting trust among untrusted collaborating parties without intermediaries [142].

A. CHALLENGES OF BC IN BPM

BPM system is complex and involves many aspects of functional requirements. It is essential for BC to incorporate other cutting-edge technologies such as IoT, Service-oriented Architecture (SoA), and Public Key Infrastructure (PKI) as a complete solution to BPM [144]. No thorough discussions were found on the challenges of this integration as complete solutions to BPM [96]. In section V, these challenges will be used for analysis benchmarks towards research works.

To facilitate our discussion, an example of transferring land titles shown in Fig. 2 is introduced to identify challenges in implementing a decentralized BP. The BP interacts with external BC systems, including (1) two public cryptocurrencies and (2) three permissioned BCs provided by municipal agencies for storing information regarding identity, tax collection, and land ownership. A trusted agency is responsible to run the BP as to settle a land transfer contract in the exchange of cryptocurrency. The execution begins with identity check (step 1) to legally conduct a land transfer between a buyer and seller. ID denotes a civic service issuing an identity to certain entity. It fulfils identity checks. Then, land transfer payment and tax payment tasks are executed in parallel (step 2). In Fig. 2, and-split and and-join connectors are syntactically used to signify parallel processes. A *payment* is initiated according to an agreed cryptocurrency to provide services for land payment (*P*) and tax payment *(PT)*. A service (*T*) from tax agency (step 3) records the payment at *tax recording task* using a BC-based tax system. The confirmation of tax payment is completed by *T* which makes a query to *PT*. A verification service (*V*) (step 4) is an internal process provided by the trusted agency that executes BP to verify the completion of tax and land payments. Land transfer (step 5) is done by *O*, a government service for land title management, where the change of ownership is recorded in a dedicated BC. Any request to BC-based services is assumed to operate under APIs. This example introduces the following challenges. Note that the development of blockchain technology for BPM is still facing various challenges, researchers have different perspectives to classify these challenges, and it is our perspective to extend our discussions on Standards for Interoperation (CH1), Trust (CH2), Confirmation of Transactions (CH3), and Irreversibility (CH4); since most surveys (Table 3) discussed the solutions to these challenges; similar classifications are used by others [174], [175], [176].

Standards for Interoperation (CH1). Proprietary protocols and implementations pose incompatibility and inconsistency in interoperation of different BC-based services. They may exhibit differences in structures, formats, functionalities, and APIs. To transact, a sender generally makes a request to a target service. Fig. 2 exemplifies the calls to services ID, PT, P, T, and O. These calls must be tailored to specific cryptocurrency or interacting services. Adopting SoA helps standardize interfaces and reduce the cost of interoperation. It is a mainstream technology; allowing users to discover and utilize online services smoothly. Leading organizations are working to integrate SoA to BC including Project 2418 by IEEE [159], 307 Working Group by ISO/TC [160], Decentralized Identity Foundation (DIF) by IBM [161], and ID2020: Digital Identity by Accenture and Microsoft [162].

Trust (CH2). When a centralized entity owns sovereignty to manage and run a BPM software, trust of the entity is traditionally established based on its reputations, business resources, and the compliances to relevant standards, laws and regulations. This context is identical to centralized marketplaces for cryptocurrency exchange. However, it was found that a dozen of exchanges were hacked in 2019, these incidents caused a total loss over 250 millions and the leak of login information over 500 thousand users [163]. These implications also apply to centralized BPM. On the other hand, decentralized BPM shares ownerships among partners. It is impossible that one entity controls executions of entire BP [26]. In such instance, trust relationship is twofold: trust of partners and trust of business owner.

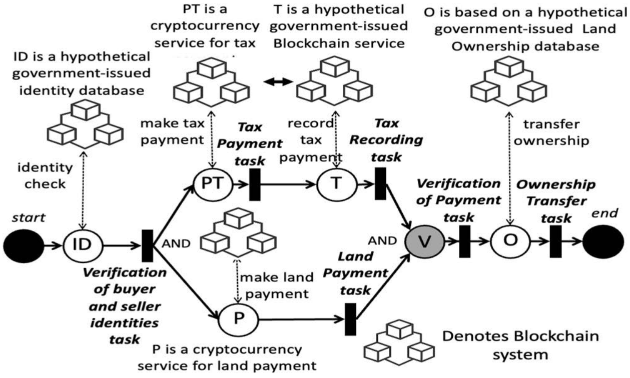

PT is a cryptocurrency service for tax
T is a hypothetical government-issued Blockchain service
O is based on a hypothetical government-issued Land Ownership database
ID is a hypothetical government-issued identity database
make tax payment
Tax Payment task
record tax payment
Tax Recording task
transfer ownership
identity check
start
ID
PT
T
AND
Verification of Payment task
Ownership Transfer task
end
V
O
AND
Land Payment task
Verification of buyer and seller identities task
make land payment
P
P is a cryptocurrency service for land payment
Denotes Blockchain system

FIGURE 2. An example of land transfer business process.

As shown in Fig. 2, one way to create trust of partners is that their services are backed by BC. However, *can a cryptocurrency for the land payment be trusted even if it is not well-known or just emerged*? The answer can be complexed when a BP requires to execute several BC-based services and some services may be selected or replaced on-the-fly [140]. Instead of trusting such services unconditionally, it is imperative to evaluate if the trust can be reaffirmed. Important factors such as BC type, number of nodes, their attributes, and mechanisms and conditions to achieve consensuses must be evaluated. When public BC involves a small number of nodes, it has a high probability that collusive parties become the majority to dominate the network. Even in Bitcoin, more than 60% computing power is controlled by six major mining pools as of 2022 [70]. In permissioned BC, the system trust is associated with the attributes of validators. Previous Facebook Diem exemplifies the issue since most validators are the companies under USA jurisdiction. Diem may gain trust from users in USA, but not applicable to those other countries with different jurisdictions. In private BC, the way of trust assurance is similar to that of a centralized system.

Increasing the number of BC-based services in BPM poses several challenges in determining trust of services; while trust is derived from evaluating and selecting services. It is unfortunate that as of 2022, neither standard nor guide of the best practice is available for trust of BC implementation. It implies that BC may be implemented inappropriately and leads to vulnerability. This explains many occurrences of ICOs frauds. Fake ICOs were accounted for an approximated loss over $500 millions in 2017 [37]. Trust issue also involves during BP execution. When trust is assured by a centralized

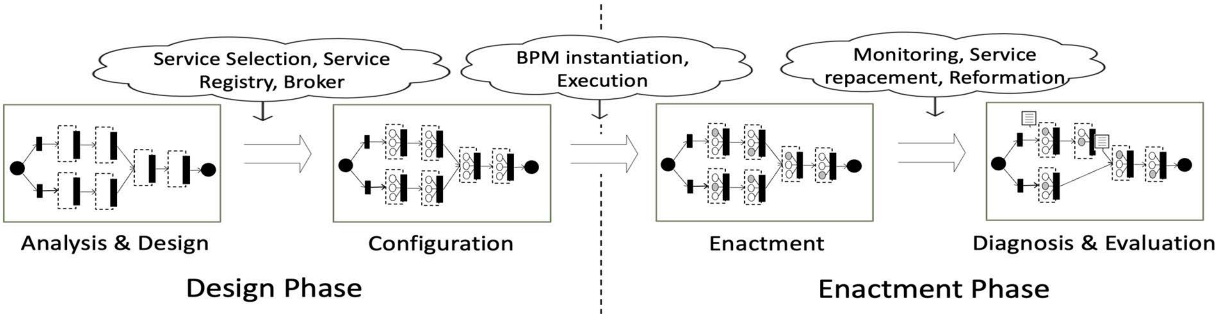

Service Selection, Service Registry, Broker
BPM instantiation, Execution
Monitoring, Service repacement, Reformation
Analysis & Design
Configuration
Enactment
Diagnosis & Evaluation
Design Phase
Enactment Phase

FIGURE 3. BPM lifecycle [139].

entity, partners must bear potential risks since they are obligated to trust the entity inevitably. The example in Fig. 2 can be configured that the execution is controlled by a trusted agency with no mechanism to validate its rightness. The task of payment verification is executed by this agency which might be corrupted. In short, trust will be assured by single authority in a centralized BPM, and it will be disseminated across partners in a decentralized BPM. Both types of BPM can tale benefits when trust is assured by BCs.

The above discussion reveals that two types of trust situate in BPM: (1) trust of services that can be assured by BC and (2) trust of an entity that execute BPs. The former is required for business to select the right services. A BC must be configured appropriately to reflect genuine trust. The latter is associated with BP executions. Apart from that, other additional concerns include (1) which nodes or domains (such as URL) are trusted to provide APIs endpoint? (2) should a requestor be part of interacting BC? if so, it is applicable to public BC rather than private or permissioned BC. Trust of APIs endpoint is usually based on reputation. Subjectivity of trust towards one service is another difficulty for service interoperations inside BPs.

Conformation of Transactions (CH3). Any BC transaction is finalized when validated and confirmed in a way that immutability is obtained. It is referred as *durability* of transaction. Confirmations are performed using consensus algorithms. Most public BCs such as Bitcoin use PoW that cannot provide exact confirmation but with probability. Private or permissioned BC uses other algorithms such as Practical Byzantine Fault Tolerance (PBFT) that synchronize consensus outputs among validators and transactions are confirmed almost immediately. This brings a big challenge to BP executions since some tasks, such as a payment for land or tax in the above example, require transaction confirmation. The way to deal with delayed or types of confirmation is not trivial since the amount of time used depends largely on consensus algorithms and BC configuration. How to deal with this issue in BPM has not been studied comprehensively despite of its importance [46]. Taking the example of making a payment in Fig. 2, it takes some time before the transaction is confirmed; however, a time-delay cannot be tolerated in some time-sensitive tasks in BP [90]. Additionally, time-delay presented in a payment using cryptocurrency can result in overdue payment [155].

Irreversibility (CH4). Immutability provided by BC becomes antagonistic to BPM when BP is implemented by smart contracts. Contracts cannot be altered once deployed on BC, enabling BP to be executed with a high degree of confidence. Since BPM usually involve several partners, it is important to have consensus on BP executions to avoid misconduct or harm. It is unfortunate that changes in BPs are common. Effective mechanisms are required to facilitate the modification of BPs dynamically [61].

Even though BC technology offers enormous opportunities in managing BPs effectively, it is relatively new, and research is needed to eliminate the challenges of BC applications to BPM. The challenges CH1-CH4 provide a reference for researchers to seek solutions in developing BPM with BC. Existing works relevant to these challenges will be discussed and evaluated in section V.

IV. BPM CHARACTERISTICS AND LIFECYCLE

Knowing BP characteristics helps readers understand the challenges of BC application to BPM and they must be addressed satisfactorily. Before proceeding to the work collections, challenges of developing a BC-based solution are associated with BP characteristics and lifecycles [139]. These will be referenced in our analysis towards research works surveyed in section V.

A. CHARACTERISTICS AND CHALLENGES OF BPM

Transient (C1) vs Persistent (C2). A transient BP refers to temporary collaboration of partners in a short period of time. Objectives can be fulfilled by one of a few executions. In contrast, collaboration in a persistent BP is more stable. Objectives can be fulfilled repeatedly by a set of processes over time. In a transient BP, services are commonly delivered by previously unknown partners; therefore, trust must be established in each collaboration. In a persistent BP, many services are provided by the same known partners where trust is determined by past experiences. BPM can be advanced to support both characteristics since business objectives can be evloved dynamically. From this perspective, BC does not directly provide a solution to BPM. However, its application offers trusted repository for Quality of Service (QoS) information, assisting service selection processes in obtaining the most suited services for BPs. Trust of a service can also be assured when backed by BC.

Dynamic (C3) vs Static (C4). BPM is facing dynamicity of BPs as changes become very common. The changes may situate in BP structures, acquired services, or sequences of executions [110]. Ramchurn et al. [112] specifically delineated the changes specifically related to partner services in open environments; since trust must be reestablished when new or previously unknown partners are introduced. Managing BP is more complicated when changes cause cascade effect on trust from the perspective of other partner services. As mentioned in CH2, even though BC provides trust of services, assessment of trust depends on several factors and subjectivity. Conventional mechanisms for security assurances widely practiced in static BPs are ineffective in assuring trust [89]. Trust models used in BPM should be scalable and flexible to accommodate changes in dynamic environments.

Formation (*C5*) *vs Enactment* (*C6*). As shown in Fig. 3, BP lifecycle involves two main stages, i.e., formation and enactment [2]. Major activities at the formation stage are process definitions for designing, planning, and creating process models, workflow compositions, tasks coordination, configurations of partner services, and process controls [25], while activities in the enactment stage aim at BP executions such as process scheduling, reselecting and recomposing services, on-the-fly service replacement, workflow reformation, runtime monitoring and analysis, and compliance of service attributes towards required QoS [137].

When BC is applied to process definition during formation, it can be a trusted repository, and it is referenced during enactment to execute corresponding processes. BP variants can be encoded by smart contracts. They ensure that a task is executed to meet the requirements given in the process definitions [52]. Transparency and auditability relate to the mutual trust in collaborations. Largely, BC contributes to BPM in two aspects: (1) it serves as a trusted repository to store information related to BPs and (2) it serves as a trusted infrastructure to execute BPs.

Centralization (C7) vs Decentralization (C8). In centralized BPs, single entity is responsible for authorizing, running, coordinating, and monitoring business operations. In decentralized BPs, they are delegated, authorized, and controlled by multiple partners. Trust of the partners has long been a major concern and BC is prominent to establish trust among untrusted partners. BC can be integrated in both centralized and decentralized BPM. Private BC is often used in a centralized BPM as a backend support, and permissioned and public BCs are suitable in a decentralized BPM [136].

With the rapid growth of services enabled by IoT and the advance of network technologies, transient BPs (C1) become popular in fulfilling quick and ad hoc functionalities, and BPM tools should be significantly advanced to deal with dynamicity (C3) effectively. It is desirable that BPM adopts decentralized controls (C8). A BC-integrated BPM will gain the benefits of immutability and validated information [157] and it is fundamental to establish trust of partner services in an open environment. On the other hand, adopting BC for trust assurance introduces difficulty in BP flexibility.

B. BPM LIFECYCLE

Fig. 3 illustrates the two main stages in BPM lifecycle. Note that researchers used different terminologies to describe these stages. BPM lifecycle is further divided into four phases [14].

Analysis & Design (L1). A BP model is created from business requirements where tasks are identified, structured, and correlated by workflow technology. BPM tool is usually featured with graphic user interfaces (GUI) to construct BP models. Many BPMs utilize formal methods and simulations with verification tools to reduce redundancies, identify and eliminate deadlocks, and detect faults. Verification activity is performed by its corresponding mathematical models, such as Petri Net, Pi calculus, or models offered by standard modeling languages such as BPEL4WS and BPMN. Outputs are conceptualized processes, documented for execution references. In decentralized BPM (C8), processes are disseminated among partners, and they are defined based on associated responsibilities. However, verification becomes very sophisticated, especially in dynamic environments (C3). Alternatively, BC alleviates this problem, providing a trusted storage in defining a BP.

Configuration (L2). Resources are allocated to tasks. Entities are eligible to allocate internal resources, or delegate to others. In service-oriented BPs, allocated resources are called *services*. Selecting services and composing them in a workflow affect overall quality. This also poses optimization challenge. Often, it is not cost effective when a BP is short-lived (C1). Functional and non-functional requirements are main factors for service selection decision. Nowadays, they are usually expressed by formal specification languages [133], [135], [140], [148] to facilitate automatic compliance checking. Decentralized BP (C8) involves rich interactions of services that can be replaced on-the-fly. Trustiness of QoS information will affect quality of service selection [145], and BC can provide a trusted repository of this information [140].

Enactment (L3). A BP instance is managed by an enactment engine that uses predefined definitions in invocating services and tasks. Typically, a partner relies on BPM to guarantee that all processes will be executed correctly as agreed. Any violation to predefined conditions will be managed satisfactorily. Assuring appropriate execution depends on the virtue of entities. From this perspective, smart contracts can be used to encode BPs [155], and it has been adopted as a core artifact to create trust of process executions [152].

Diagnosis and Evaluation (L4). A BP is executed and monitored. Outcomes are analyzed and evaluated against predefined functional requirements [95]. In [22], 104 cases of BPs were analyzed to retrieve specification patterns for QoS evaluation of service-based applications.

Apart from that, flexibility and adaptability are emerging as significant measures in BPM. In open environments, services in a BP are often reconfigured or updated to address real-time changes. For example, when better services are found, a BP may be updated, and this affects activities in other phases of the lifecycle. The role of BC in assuring trust of service is different from that at configuration phase (L2) where BC is used to provide trust of QoS information; here, activities and processes are critical since they are relevant to the updates of a BP. Smart contracts can be implemented as monitoring processes [140].

It is seen that BC can be applied, directly or indirectly, in all phases of the lifecycle. BC provides trust of process definitions (L1), executions (L3), and supports BPM with the monitoring information and processes for later analysis (L4). Furthermore, BC also provides trusted QoS information for selection and composition of services (L2).

V. BC IN BPM

Numerous efforts found in literature have been increased drastically in the past few years. Its ability to provide trustless architecture makes it a pivotal research interest for collaborative BPs. With reference to BC challenges (CH1-CH4), BPM characteristics (C1-C8) and lifecycle (L1-L4) discussed in section III and IV as benchmarks, they enable us to analyze research advancements in a systematic way. The following topics are classified into ten major categories. The analysis result is tabulated in Table 3.

A. BC-ASSISTED BP EXECUTIONS

BC in the early date is deemed as a supporting component for securing BPs. The early adoption considers it as a recording mechanism. Later, smart contracts create an innovative stream on securing processes. Primary studies have been conducted by exploiting these benefits. This section surveys and discusses the use of BC as a supporting component to BP modeling and execution.

A work in progress by Cen et al. [30] suggests a BC-based framework to secure inter-organizational BPs by ensuring that external processes are managed and secured by BC. BC is used as a shared database, where data regarding execution states, process transition, and outputs, are kept. Data elements are predefined from mutual agreement from partners. They consist of milestones as check points to confirm execution sequences and outputs. Process controls can be verified simply by checking sequence of transactions. Flexibility is a major concern because each change requires re-synchronization as a new version. This approach faces significant time-consumption and is expensive since it is not generic to serve all agreement processes; collaboration is done on a case-by-case basis. The implementation is realized by Fang et al. [47]. They first pinpoint the problem of trust from the reliance on a centralized party, accounted for coordinating and governing controls over executions. Typically, a trusted third party is used to provide and warrant trusted environments. To mitigate this, the authors present a novel architecture called Workflow Enactment Service (WES), allowing multiple BPs interoperate together via a common BC. They develop in-house protocols and interfaces to smooth communications. Smart contracts deliver synchronization of BP states implemented in Hyperledger Fabric. One major weakness is the lack of details pertaining to how methods are applied to process models [53]. Later, Alves et al. [9] close the gap by devising a technique and integrating WES into Camunda running on Hyperledger Fabric. The workable artifact is demonstrated through a use case of geophysics data acquisition processes. However, only practicality is evaluated while in-depth analysis towards strengths and weaknesses is missing.

Chang et al. [32] identify the aspects of BPs where smart contracts can be useful. A supply chain scenario is chosen as a running example, where important aspects are extracted. Like the work mentioned in [30], the first aspect is to keep track of transaction regarding process status. BC is used to record provenance, providing access to partners. Another aspect is described in payment processes. Using smart contracts to implement decision logics can avoid payment escrows. They provide operational controls among partners without intermediary. These two aspects can be combined to automate BPs with the benefit of time and cost reduction. However, the work does not provide implementation and evaluation details, diminishing its creditability. To exemplify this, Panduwinata and Yugopuspito [108] demonstrate how a simple BP of a parking system modeled by BPMN can be mapped into structural elements of Hyperledger Composer. The work confirms that parts of BPs can be reengineered by the platform. On the downsides, the mapping process between BPs and Hyperledger Composer elements is manually done. Only a simple scenario was tested, where, in fact, the real-world business processes are far more complicated than the simple case in Fig. 2. Therefore, using BC in this example does not reflect sufficient capabilities of BC. Many aspects, such as ones addressed previously by Chang et al. [32], are left to be studied. Significant progress is required towards a general framework with useful patterns. Nonetheless, the proposed technique preserves its value and can be improved further. By the way, it is important to note that Hyperledger Composer has been deprecated since 2019.

BC applications have also been studied in the academic domain. ProChain [33] is developed for provenance sharing of scientific workflows. BC replaces traditional centralized provenance architecture to deliver better trustworthiness, reliabilities, and efficiency. The block design and data structure can be adopted to store fragments of BPMN logics [53]. However, it lacks implementation details. Fernando et al. [49] argue that data stored in ProChain is data product rather than provenance. They propose SciBlock on Ethereum with a new data structure to additionally store temporal information, which is important to identify outdated results. Hence, researchers can determine trustworthiness of result reproduction. In the eScience domain, two lines of research are envisioned [69]. (1) BC is used to store provenance data, and (2) smart contracts enable the adaptation of workflow choreographies, which are an essential part of collaborations.

In summary, various attempts tailor BC as a component to support different aspects of BP executions, which are largely classified into two main usages: (1) BC as a trusted storage to keep track of BP execution states and outputs, and (2) smart contracts as a mechanism to encode conditional logics and parts of execution controls.

B. BC AS AN INFRASTRUCTURE FOR BP EXECUTIONS

BC can be an infrastructure for enforcing executions of BPs. The fundamental idea is to code process logics into smart contracts. A trustworthy environment [96] is created among untrusted partners by ascertaining that pre-defined process models will be followed correctly (CH2). The infrastructure can be divided into three main aspects: process-centric paradigm, data-centric paradigm, and smart contract-less.

Process-centric paradigm. The early vision is suggested by Mendling [94] describing potentials and challenges of BP control flow implementation in BC infrastructure. It lays a foundation for subsequent developments. This original work merely provides limited insights of how an architecture and artifacts are built and combined, rather, indicates a guidance for future development. Weber et al. [152] realized the idea by illustrating the possibility to execute BPs on a Ethereum at the first time. BC is used in two aspects: (1) as active mediator for BP executions where actual data processing is performed by smart contracts, such that only conforming cases are allowed, and (2) as a choreography monitoring to log message exchange. A special set of smart contracts is developed to facilitate and control the creation of BP contract instances. To ease the transition from the design to implementation, an automated translation artifact from BPMN to smart contracts is devised. However, the paper does not entail technical elements. They are referred to the technical report [151]. This work also faces a scalability issue since the developed smart contracts are not generic, which need to be generated per process model. In response to this, Sturm et al. [128] develop a generic smart contract set based on a technique that imitates state-firing transition. Tasks are considered states and will be fired when requirements are fulfilled. The smart contracts serve as a scaffold and are independent from any process model. The prototype is proposed on Camunda to illustrate its practicality. García-Bañuelos et al. [52] improve the works by aiming at resource optimization. The main contribution is a compilation engine. It starts from detouring the translation from BPMN into Petri Net, in which verification algorithms are applied. Hence, throughput rate and runtime components are optimized. Then, the second compilation translates Petri Net into Solidity. Another compilation technique is proposed by Nakamura et al. [106] by translating BPs modeled by BPMN into a statechart. Its main purpose is to relieve the bottleneck of BC operations by optimizing communications between partners and BC. The main advantage of state-chart is that it can be automatically translated into smart contracts. The prototype is implemented on Hyperledger Fabric. However, the current progress only covers basic elements. More complex control flow patterns found in [119] are left untouched.

Caterpillar [84] is considered a complete generic framework to execute BPs. Its core process engine is implemented following the concept by Weber et al. [152]. This approach relies on BPMN standards and abstracts control flows into Solidity. It is not intended to replace current BPMs, rather featuring them with BC capability. More details are found in the extended version by the same authors focusing on building BPMN-to-Solidity compiler [85]. The proposed artifact is the engine where states and tasks are stored and executed. Unfortunately, technical elements are not satisfactorily explained. The complier functions regarding how the system works is not included. Lorikeet [130] is another practical framework like Caterpillar in the aspect of transaction technique. In addition to storing process logics, Lorikeet implements registries as trusted storage of asset. Both Caterpillar and Lorikeet provide high-level languages to facilitate users to obtain Solidity code from BPMN. However, how the code is executed is lacking [53]. Compared to Caterpillar, Lorikeet is equipped with a feature of partner selection (L2), which is superior in terms of access control, while Caterpillar supports greater BPMN elements [34]. At present, they are considered the fully-fledged process-centric frameworks with ability to manage subprocesses, multiple-instance activities, and event handlers. BP executions are deployed entirely on a BC. Regardless of platforms, these works altogether suffer from flexibility, rooted from a fixed implementation focusing on translation of orchestration diagram (L1) into smart contracts, inhibiting flexibility or adaptability during runtime (L3). Adaptability is complex as it contradicts with BC basic principle of irreversibility (CH4). In addition, despite implementing on BC, there is no guarantee of misconduct if a BC is owned by single or some colluded partners that can direct the consensus outcomes.

Another line of research is originated by speculating that flexibility problem arises from excessive desire on strict controls of process executions. However, there is an increasing demand for flexibility as partners are unnecessary to be restricted regarding how they execute processes. Flexibility is needed for BPs to adapt to change. Most BPMs are flow-based with various degrees of restricted controls flows. They are confronting a problem when dealing with changes. To mitigate this, a declarative approach is introduced [1]. The core concept suggests that any execution is possible as long as it will not violate constraints. Partners can autonomously operate on their private implementations to execute processes. The relaxation of restrictions means the higher degree of flexibility. Typically, the constraints are enforced by a trusted third party to verify actions. BC can eliminate this as a declarative model can also be coded by smart contracts, as the proof of execution correctness (CH2). Madsen et al. [88] realize this concept on Ethereum. The methodology relies on Dynamic Condition Response (DCR) graph to express valid conditions suggesting restricted constraints. Any violation indicates fault. The DCR model is translated into smart contracts, whilst the sequence of executions is recorded on BC for traceability.

Data-centric paradigm. The data-centric paradigm is getting much attention along with the popularity of big data. In BP collaborations, smart contracts are an event-driven artifact for data processing in a distributed manner. Hull et al. [67], [68] envision that this paradigm can outline a foundation for BP executions on BC. Conceptually, BC is viewed as data anchor, where data to be recorded is controlled by smart contracts. This work follows the logic separation principle, presenting Business Collaboration Language (BCL) to separate the low-level smart contract coding from business users. B-MERODE [11] follows such approach by introducing interleaved constraints for pre and post conditions to indicate the completion of tasks. This approach dismisses the implementation of control flows. Instead, it stresses on BC artifact to keep the state of data processing. The approach guarantees trust of milestones, suggesting the set of states based on data updates according to pre and post conditions. It is an upgraded version of the work by Cen et al. [30]. However, only methodology is presented while important details such as implementation and evaluation are missing. It lacks foundations regarding semantics, architecture design, and context of applications. While the data-centric approach utilizes BC for trust of data, the guarantee on execution correctness is not fully completed (CH3) because role-binding and process logics are not controlled by BC. Unlike the process-centric paradigm, flexibility is casually claimed since BP logics are by design not rigidified by smart contracts.

In either process- or data-centric approach, analyzed by [127], the works implemented on Ethereum are facing the inherent problems associated with public BC. The evaluation demonstrates inefficiency in terms of performance. Adhering to Ethereum incurs gas consumption implying additional cost in running BPs. Furthermore, to interact with external entities, special APIs must be added to original Solidity with the special concern of being deterministic. Problems related to Ethereum are further discussed in [109]. Other BC options like Hyperledger Fabric or Quorum can be considered to eliminate these shortcomings.

Smart contract-less. Much of the research focuses exclusively on the use of smart contracts to coordinate processes and control violations. The weakness to the approach is the *state distinction,* (discussed later in section VI). The works introduced by Härer [58] and Evermann and Kim [43] take an alternative way with the belief that smart contracts are not the only way to enable trust of BP executions. They notice that the translation of BPs into smart contracts incur unnecessary cost and error prone. For instance, all parameters, sequence, states, and controls of logics must be completely predefined. Instead of implementing a model-specific engine, a process model can be done on the fly based on a mutual agreed shared ledger. This modeling technique is first introduced by Cen et al. [30] and its concept is realized by Härer [58]. He opts to use contract-based BP modeling where the model is formed by partners. Thus, a pre-defined model is inessential. The role of BC is to track workflow instances, assuring the state of work and indicating next valid activities, such that integrity of model execution is preserved. Evermann and Kim [43] advance the work by implementing a workflow engine directly on BC and introducing standard interfaces to interact with BC infrastructure. The implications between traditional and BC-based WfMS are identified through experiments with several design choices. In short, the concept utilizes BC solely as a recording mechanism while the current state and next valid activities are determined by partners. Flexibility is achieved as internal processes are preserved from others. The main concept is simply to shift the validation load from smart contracts programming to external agreement. Despite cost reduction, the weakness is found on the excessive agreement operations in which every next activity requires validation.

In summary, there is no single BC system that fits all use cases [45]. Types of BC and their trade-off [136] must be determined for suitability. Choice of BC types needs further investigation for suitable adoption in different contexts. The issue will be discussed in detail in section VI.

C. FLEXIBILITY OF BPS RUNNING ON BC

Flexibility becomes a key property to enable adaptability towards ever-changing requirements. BPs are evolving progressively [73], which require adaptations. These changes are classified in two ways: (1) changes in requirements regarding the eligibility or quantification of partner services, which may result in service replacement, and (2) changes in BPs to adapt to newly emerging needs. However, early BPMs assume that process definitions are mostly stable (C3), where coping with flexibility is a long-standing issue. It is more complex if BPs are deployed on BC. It is contradicting irreversibility (CH4).

To react to this, L´opez-Pintado et al. [81] promote flexibility specific to the change of partners, where dynamic role binding and unbinding activities situate during runtime (L3). Binding Policy Specification Language (PSL) provides high-level notations to govern changes. The extended version [83] includes process binding to tackle changes of BP structure. It enables (1) dynamic binding of partners and (2) empowers partners the control of sub-process and execution pathways. The changes are accomplished by off-chain consensus, such that partners must validate each update before deployed. Such update is regulated by PSL to restrict how a valid decision is determined. For instance, a policy may state that change of suppliers requires at least 50% from partners votes. This approach is prototyped in Caterpillar. As a result, the problem of irreversibility (CH4) is lifted in two aspects: (1) process adaptation to deviate executions, and (2) evolution for the permanent changes to entire BPs. Similar technique is employed by Klinger and Bodendorf [72]. Instead of using orchestration diagram, they rely on BPMN collaboration diagrams to avoid additional transformations. Subscription service is created to mediate communications between on-chain to off-chain systems. Changes are controlled by static smart contracts that regulate how changes can be done [155]. This concept is equivalent to PSL, but it is done entirely on-chain. However, applicability is a main drawback. The need of running consensus for each update is time-consuming and inhibits agility. In addition, PSL incurs additional overheads. They perhaps are not exhaustive enough to cover unexpected changes. Thus, it is inconvenient for business partners to operate in dynamic BPs (C3).

The previous works deploy a complier concept to transform BPs into smart contracts. This causes process instance mostly inadaptable at runtime without redeployment. To mitigate this, Lopez-Pintado et al. [82] enhance Caterpillar with an interpreter and dynamic data structure. Like [128], common operations are hard-coded into smart contracts based on BPMN. Separated smart contracts allow updates to process instances. Another advantage is that the technique can address entire process variants. However, the adaptation ability lessens the tamper-proof as it allows changes to be made at runtime [53] (L3). A solution presented by Evermann and Kim [43] uses a technique by generating all possible outputs and changes are simply done by relocating BC heads. However, it suffers resource waste as all combinations must be pre-calculated.

We believe that the most viable and workable approach is presented by Klinger et al. [73]. Their strategy is to decouple the versioning of process state and business logics into separate smart contracts as to avoid changes made directly to smart contracts. They present registry pattern that is written with current execution version while proxy pattern is used to enforce BP logics. The change of partners and processes are done through separate smart contracts that implement democratized mechanisms such as voting. Depending on situations, the contracts are a static component representing logics of how changes can be achieved. Any change will update the versioning state, reflecting the latest version in use. Evaluations of this work are reported with cost, overhead, and complexity.

D. RUNTIME MONITORING AND TRACEABILITY OF BPS USING BC

In decentralized BPMs (C8), parts of processes are distributed among partners. Usually, no single entity has full control as partners are independent, and assigned processes are locally executed. Contracts are used to describe functional deliverables and QoS of partners [107]. Specification languages are extensively used as a markup in the contracts [148]. The language specifies what properties of partners to be evaluated during configurations (L2) and monitored during runtime (L3). Unlike centralized BPMs where information is readily available to owner, decentralized BPMs face a major challenge of end-to-end visibility towards the correctness of executions [138]. This information is imperative when resources or data belonging to an entity are manipulated by others [100]. To achieve this, the information must be traceable in which it must be documented with trust [19]. This information is helpful to verify execution correctness as well as pinpoint problem areas. In the business domain, the verification focuses on the compliance of functional and non-functional requirements, often specified by Service Level Agreement (SLA) [135]. Most traditional techniques use a centralized system for these activities. They experience high overheads and recorded information cannot be trusted [109].

"Traceability for decentralized business processes implies the capability of tracking the status of ongoing instances and reconstructing the history of its execution [35]*"*. It is often constructed from monitoring information recorded during runtime. However, the lack of trust between partners hinders accurate and sufficient information. This issue is viewed as an initial benefit that BC offers. The basic idea is that BC provides a trusted storage for recording and retrieving this information. A survey exists on this topic [36], but it is not comprehensive. Besides, we refer the criteria proposed in [113] with four major classifications of monitoring objectives as reference: (M1) Event data logging, (M2) Activity monitoring and runtime performance analysis, (M3) Conformance checking, and (M4) Compliance checking.

The work towards this topic is first proposed by Weber et al. [152] and later becomes a part of Caterpillar. A monitoring artifact called C-monitor is devised as a broker to handle message passing, such that message exchange among partners are recorded (M1). Focusing on M1 and M2, Prybila et al. [109] employ Bitcoin infrastructure to store monitoring information (L4) for every step of executions. Bitcoin transactions are modified with new data structure to capture state and information of BP executions. The obvious benefit of Bitcoin is practicality, but it faces scalability and adaptability issues. Additionally, Bitcoin has an unfavorable reputation for energy consumption with slow and unreliable time for transaction confirmation (CH3) [50]. Although this work emphasizes on M1 and M2, M3 can also be satisfied with extension as information neccesary for conformance checking are available and shared across partners. The work by Di Ciccio et al. [35] also enhance a traceability mechanism to Caterpillar. It depends on the assumption that entire BP executions run on BC. BC is viewed as an infrastructure to store all execution information. To reconstruct process instance, reverse-engineering is performed on Ethereum hash codes. This information satisfies M2 and M4. A real case in the pharmaceutical domain is presented to demonstrate applicability. However, some weaknesses are identified. Performance evaluation is missing. The information needed for traceability is limited within the connection of transactions as to identify and regenerate process instances with associated cost and the entity that performs a task. Meroni and Plebani [99] analyze the possibility of using an artifact-driven approach for BP monitoring based on BC. They introduce a completely different approach by using IoT devices for monitoring operations. The completion of activities is determined by peer IoT devices regarding pre and post conditions of an interested device. BC decentralization is suitable in establishing trust in this setting. This way, partners participating in

BPs are free from monitoring activities. The main advantage is that all monitoring processes can be made automated. The only implementations are the notification mechanism to peer devices and BC to store information. Only one important assumption is that IoT devices are smart and high in capacities to handle BC operations. The authors further developed a fully BC-based platform as a trusted monitoring solution for BP modeled by BPMN [100]. An IoT-based case example is demonstrated to collect event data (M1) by modifying a router module to send data to BC. They solve the ordering problem in BC which does not guarantee that monitoring information is stored in a chronological order. During data gathering, the process of data collection is encoded in Solidity, such that it is reliable and cannot be compromised. Technically, all approaches use BC for information logging. Therefore, they are transparent and auditable, enabling a valid source for anomaly detection.

Related work by Haarmann et al. [57] state that decisions taken by interacting partners are also essential to solve many conflicting situations. Consistency is paramount to avoid repudiation. Inputs for calculation must be trusted where BC can affirm this. It provides a reliable and verifiable basis for conflict resolution. The foundation of this work is based on Decision Model and Notation (DMN) and uses BC to capture decision structure and decision logics (M1). We believe that the decision logging will yield greater benefits by integrating to the work by L´opez-Pintado et al. [81], such that the decision of binding and unbinding entities to roles and subprocesses are logged. It can be later used as a proof of changes. However, the prototype is implemented on Ethereum without justifications. Thus, cost specific to gas in this platform, which is quantitatively evaluated, is of little value. Silva et al. [124] encompass human interactions in an organization setting using DEMO concept [38] to design the existent transaction types with social and communicative aspects. DEMO is integrated with Hyperledger Composer (obsoleted since 2019) to define design abstractions for BC-based BP executions. They design and implement a prototype to validate the model in a food supply chain scenario, which represents several complex processes and dynamic partners. However, model assessment towards threats such as information distortion is not given.

In summary, most of the existing efforts focus primarily on M1. Runtime monitoring employs BC as a trusted storage to track information and decision-making. BP status and data retrieved from executions are validated. Trusting the data is rooted from trusting monitoring processes. Fortunately, existing works provide trust by encoding the processes into smart contracts. Apart from that, the correctness of BP executions must also be monitored by partners, which can be indirectly accomplished. Process mining technique is employed to reconstruct processes from attributes extracted from event data resided in BC [35]. For instance, the order of executions can be regenerated from BC transactions. M2, M3, and M4 are achievable by using this technique. A few interesting works on process mining in BC have been reported, which will be devoted in the next subsection. Runtime monitoring research is advancing towards automation with high-level analysis for M2, M3, and M4. Data privacy and openness are two issues to be aware of.

E. PROCESS MINING ON BC FOR BPS

Process mining is a common feature in BPM discipline. Its role is to diagnose and reaffirm compliance and performance of BPs, which subsequently improves overall operations. The goals are divided in three aspects [74]. (1) process discovery for analysis (M1, M2), (2) compliance and conformance checking (M3, M4), and (3) model enhancement driven by the analysis. The first aspect inspects if executions are correct, the second determines that the processes conform with specifications, and the last makes use of analysis results to improve BPs.

Process mining on BC first appears in the reconstruction of transaction flows in Bitcoin. GraphSense [60] is an analysis tool to track user behaviors from wallets and transactions to construct graph-based representation of the flows. However, it is limited within cryptocurrency. In collaborative BPs, extracting information for analysis is extremely difficult as BC transactions are not originally designed for this. To alleviate, Duchmann and Koschmider [41] introduce an approach to analyze the logic flows of smart contracts as to ascertain that the process logics on BC is executed as defined,. The flow is constructed by mining event logs (M1) to obtain necessary information to regenerate the sequence of executions. Then, it is abstracted by Petri Net for verification purpose. The approach is implemented on Hyperledger Composer (obsoleted 2019) with remarks that Ethereum is not suitable for process mining as it is not naturally built to support BPs. However, Ethereum is currently the most attractive platform as evident by a growing number of BP research, despite many problems. For example, the transaction order does not assert chronological processes, logs are minimally stored to reduce size and cost which often contain insufficient information, timestamp is approximated as it is defined at the block level, and data structure is inconsistent in each transaction. To solve this, Klinkmuller et al. [74] propose a framework to facilitate the process extraction from transactions to turn into Ethereum Decentralized Application (DApp) event logs. The specification and meta-model of event data and format to be logged is manifested, representing clear constraints on data types, and providing consistency, such that the log generated by BC can be enhanced for process mining. Muhlberger et al. [104] broaden the work by converting transactions into event logs conforming to IEEE Extensible Event Stream (XES) standard. This is significant in a way that XES is used as a main format in process analytics toolkits [116]. These works can be nicely integrated with Caterpillar or Lorikeet. However, the remark is given regarding the contributions that are limited within the Ethereum platform.

In summary, most of process mining attempts employ Ethereum platform. In reality, interoperation of BPs often interacts with other BCs [147], which can be any of public, permissioned, or private type. Future research may direct to process mining standards and practices regardless of BC type. It also solves traceability challenges regarding insufficient information for process reconstruction. Yet, the research in compliance and conformance checking (M3, M4) on BC has not been found. This paves an avenue for future investigation.

F. SMART CONTRACT VERIFICATION AND OPTIMIZATION FOR BPS

Ensuring the correctness of smart contracts is essential to prevent unexpected behaviors of BPs [91]. Difficulties arise for BPMs to translate BPs into smart contracts. Programing smart contracts usually contains errors [121]. Consequently, it leads to BP discontinuity, diminishing trust, and economic losses. Fixing programming or logic errors later on will be costly [132] owing to the irreversibility (CH4). The problem is escalated since there are various BC systems, with difference in foundation, programming languages, features, and characteristics. Smart contracts on some platforms such as Ethereum consume gas to fuel executions, which induces extra lines of code compared to other BCs. Discrepancy between pre and post translation is costly to fix and often detected at the time when BP has already been enacted (L3).

Although research in smart contract verification has been conducted [59] and a survey article is available towards comparison among various verification methods [8], they are too general and only a few proposals have been found to tackle the issue in the context of BP. Amani et al. [10] implement an extension on Ethereum Virtual Machine (EVM) with formal verification to detect errors, covering syntax and typechecking when a BP model is compiled. ContractLarva [15] is a runtime verification tool to reason about smart contracts and predefined BP definitions. The extended works by Ellul and Pace [42] and Yu et al. [156] demonstrate how standard techniques from runtime verification can also be applied. Using the same approach, Azzopardi et al. [16] employ deontic logic as a formal specification tool to this verification. It ensures that BP execution is not diverged from intended behaviors. However, these approaches require an independent set of smart contracts to implement specifications. The process mining-based approach by Duchmann and Koschmider [41], reported previously, can avoid this overhead, but it introduces greater delay to detect anomalies. A more promising approach to verify Solidity employs Finite Stat Machine (FSM). Smart contracts must be assured that vulnerabilities are eliminated as much as possible since its logics are difficult and costly to be altered after deployed. FsolidM framework [91] is developed specifically to solve common vulnerabilities studied by Atzei et al. [13] and introduces design patterns in Ethereum. To simplify the processes, a GUI enables developer to model FSM-based BPs that will be complied into smart contracts automatically. Using FSM has several advantages. It has long been practiced where plenty of efficient reasoning tools are available. However, state explosion related to FSM is not discussed and using Ethereum can be costly to run BPs. The greater number of smart contracts, the higher cost of executions. Reducing the cost by optimizing smart contracts are essential. Hu et al. [64] employ modeling techniques by translating BPMN into Petri Net and present a mapping technique to reduce duplications and enable tasks fusion. To prove this, algorithms are verified that the reduced version behaves identically to the original one. Its effectiveness is illustrated through an experiment suggesting that gas, depending on the complex of tasks, can be saved maximally almost 20% compared to traditional implementations.

In summary, generating optimized error-free smart contracts is challenging. Implementing smart contracts requires deep knowledge. It is very time-consuming and error prone. In this regard, errors from the translation of BP model into smart contracts can be diminished by verification techniques. Aspects of verification can be done during compilation and runtime. It is worth to note that formal methods and specifications should be incorporated by translating a BP model into mathematical notations such as Petri Net or FSM, where rigorous verification and reasoning tools are prevalent. The goal is to eliminate discrepancies between expected behaviors and smart contract executions.

G. BC INTEROPERABILITY IN BPS

Opportunities provided by BC allow modeling secure interactions among BPs and partners. However, many technical problems and concerns of integrating BC into BPMs exist. Previously, the main challenge that BPM must deal with is that proprietary protocols created by different in-house IT systems are mostly incompatible (CH1), which makes them difficult to interoperate. This issue also applies to BC systems. Within the context of BC, uncertainty in transaction confirmation delay (CH3) poses critical issue for time-critical BPs. Moreover, partners may host their BC systems with different types. These become big obstacles for organizations to interoperate with multiple partners that may implement private, permissioned, or public BCs. The works found in this area attempt to solve interoperation challenges as we have elaborated in the BP example in Fig. 2.

In this regard, there are two studies that survey BC interoperability found in [61] focusing on IoT and [80] for general BC interoperability. Compared to the former, we differentiate our survey introducing broader coverage than IoT. The later includes articles conducted for BC-to-BC interoperation at the protocol level, which is inefficient in the context of BPs. Many studies are working towards schemes of BC interoperations.

The early mechanisms attempt to establish communication between BCs, which can be integrated into BPM. The first mechanism is a notary scheme suggested by Buterin [28] to facilitate interaction between BC-to-BC. A group of trusted nodes are found to construct a special type of BC addresses, called a multisig (multi-signature). These nodes must be part of both interacting BCs to govern processes of information exchange. Although it is intended to ease asset exchange between two BCs, this also holds the transitive property as an asset in chain X can be exchanged to chain Z through chain Y (see Fig. 4). The second scheme employs a technique called sidechain, which is a dedicated BC serving as a medium between interconnected BCs. Sidechain is authorized to access and validate transactions of interconnecting BCs. This scheme is more efficient than the notary scheme as it eliminates the one-to-one implementation requirement. Interaction among several BCs may involve one or multiple sidechains. Technically, sidechain relies on message passing where the delay or types (whether probabilistic or realistic) of transaction confirmation (CH3) is well-aware. An asset will be released only if all involved transactions are considered confirmed. Like the notary scheme, sidechain needs trusted nodes to operate. Fig. 5 conveys the overview of a sidechain architecture, involving multiple sidechains, and communications to main BCs. Blockstream pegged sidechain [18] exemplifies this technique. To promote flexibility and scalability, a zone and hub architecture is introduced, which is very effective in a large-scale environment such as IoT [143].

Initiatives were found specifically to study BC interoperability. For example, Blockchain Interoperability Alliance (BIA) was found in 2018 to increase awareness, identify the needs of interoperation, and develop methods for collaborations of independent BCs. It focuses on spontaneous interoperation and asset exchanges during BP executions [164]. Note that interoperation could be possible when businesses hosted in different platforms can interact with each other [27]. With the same acronym, Blockchain Industrial Alliance (BIA) was initialized in 2020 [165] and Ethereum DApp was proposed as a platform for decentralized BPM, in which BPs can be interacted and integrated with BCs.

Although BC interoperations are extensively realized in the world of cyptocurrencies, they can be extended to the BP domain. However, feasibility becomes a roadblock. The interoperations in BPs are dynamic; while existing works on BC interoperations have shown several limitations: (1) supporting interoperations only at the protocol level, (2) confining the implementations within cryptocurrencies, (3) focusing on BC-to-BC interoperations that was costly and impractical to be applied in a large scale such as in dynamic BPs, and (4) inapplicable to semantic interoperation while solutions to such need are desirable. We notice that interoperations should be committed at the semantics level with the help of semantical ontology to interoperate among various partners. In this vein, there are two main mechanisms have been reported so far.

The first mechanism utilizes smart contracts as a medium to communicate with other BCs from various partners. To enhance existing BPMs to possess an intuitive model to interacti with BCs, channels for communications with BCs need to be initiated. BlockME [44] equips BPMs with specialized models and execution features to solve uncertainties related to BC access and operations, e.g., transaction confirmation (CH3). The design principle makes BlockME technology-agnostic and become a unified middleware for communicating with several BCs and their variants. However, its status is bound for asset exchange. This work makes use of APIs, allowing external entities to execute operations in a BC. For example, an API is triggered to notify partners when a specific transaction is confirmed. All operations are done inside a Blockchain Access Layer (BAL), where different characteristics associated with each BC are identified and resolved before operates. One side benefit is that it eases developers to model BPs on BC without deep knowledge. BlockME2 [45] extends the original BlockME by featuring it with standard-compliant BPMN. With the help of BAL, BlockME2 facilitates translation of BPs modeled by BPMN into BC. Smart contract functions are agglomerated from collaborating partners that implement smart contract enabled BCs. They are composed to smooth the translation from BPMN logics. Its current progress integrates well-known BCs like Bitcoin, Ethereum, and Hyperledger Fabric. One technical downside is that the setting of the work series assumes all involved partners' BCs are trusted. However, indicated in CH2, it is risky to be assumed. Comparison to other state-of-the-art approaches towards structural and temporal rationales is not provided in these series [53].

We believe that BC potentials can be fully unveiled with combinations of cutting-edge technologies. Technologies behind BC are not new. The applications of them in a novel way make BC valuable. To promote BP interoperations, traditional organizations try to be opened by implementing APIs as a main channel to access its in-house IT systems or Clouds. Most APIs are proprietary. The difference in implementation increases overhead for interoperations. It also inhibits dynamicity in which each change requires full understanding of new APIs. The same situation holds when interoperating with BCs. For example, interoperating with BlockME2 must understand all smart contract functions from involved BCs. Heterogeneity is the main difficulty. For instance, making payment tasks as shown in Fig. 2 implies that the agency that runs the BP must develop API requests, being tailored to different cryptocurrencies. This situation imitates the history of SoA. Its goal is to standardize communication among distributed web services with unified architecture. BlockME can be abstracted by SoA to facilitate collaboration with other BCs smoothly. Like web services, BCs can be viewed as services and can take benefit from SoA (CH1). Viriyasitavat et al. [144] and Dinh et al. [39] address this outlook. Authors in [142] further elaborate the perspective of design elements, which are synthesized and surveyed from several use cases regarding SoA-integrated BC in IoT ecosystem. The finding reveals most existing studies are segmented and diversified. A unified solution must be seriously investigated. In this focus, a conceptual architecture with SoA-integrated BC towards seamless interoperation is proposed in Fig. 6 [147]. A broker concept is adopted. It is selected from high-capacity IoT devices serving as API gateways. Encapsulating with SoA, collaborating partners can expose their services implemented by smart contracts, which are wrapped by SoA standard interfaces. Services can be indirectly invoked by APIs and associated smart contracts will be subsequently invoked. Like BlockME, the challenge of transaction confirmation is addressed (CH3). They suggest that permissioned BC with zone and hub architecture [143] is well-suited for BPM to deal with the large scale of BC-based IoT services. However, high volume of data generated by IoT devices becomes a roadblock whenever BC is used as storage. To increase performance, Xu and Viriyasitavat [155] address the issue by introducing BC tree structure [126] to handle a high rate of transactions. Together with a high-level consensus, it achieves better flexibility while trust is retained. Unlike popular consensus algorithms such as PoW and PBFT, this type of consensus is done at the smart contract level. The technique provides flexibility in activities such as specifying partners and task assignment. Partners can specify mechanism together for consensus to run the activities. Still, the base consensus of BC is left intact. It operates as normal. Tasks that require high degree of trust can be configured with considerate partners to run the high-level consensus before execution. Partner selection is done based on agreement from all partners in BP during configuration (L2).

In summary, BPMs tend to interact with disparate types of services including web services, IoT, in-house systems, Cloud, Edge computing, and BCs. In this context, IoT services are growing exponentially and many of them are backed by BCs to promote trust. The prevalence of services demands communication standards to lubricate interoperations in BPs. As mentioned, this issue has been responded by leading organizations (see CH1). BlockME research series [44], [45] can be complement with the works regarding SoA-integrated BC [142], [147], [155], paving the way for future studies.

As mentioned in CH2, another aspect left to be solved is the trust of BC systems themselves. Neither standards nor best practices are available to guide BC implementation in a way that can be trusted. Business must bear its own risk if BCs in use are implemented inappropriately. Trust of BCs depends on many factors, for example, the number and types of validators or nodes, methods for encryption and digital signature, consensus algorithms in use, etc. Especially, compared to public BC, permissioned and private BC are considered less trusted as a minority involves in consensus [127]. The issue is leveraged in accordance with the growth of BC-based services. We notice that PKI can be a suitable solution [142] for trust establishment. Identical to certificate of domain names, service attributes can be extended to cover BC as a proof of implementations. The primary advantage is that PKI has already been proven. We can imitate the processes for verifying domain names, extend them to cover BC implementation and endorsement [147], and make use of already-established infrastructure. However, strategies to establish and promise trust by endorsing BC as a service need further studies. The right balance between cost and benefits must be evaluated attentively.

H. RESOURCE-AWARE BPS ON BC

So far, research found in BC-enabled trust of collaborative BPs attempts to find practical solutions to bridge the world of BPs and BC as a trusted infrastructure for executions. Most attempts relieve the pain-point related to automated translation from process models during choreography (L1) into BC platform for enactment (L3). Several extended works enhance flexibility at runtime (L4). During this transition, one important space to be explored falls within configuration phase (L2). The core activity is to allocate resources for tasks. If the resources are SoA-encapsulated, the term service allocation is commonly used. This activity is tightly coupled with access control where entities that provide resources are authenticated and authorized for their assigned tasks. Briefly, the main purpose is to answer the question of who is permitted to execute which tasks or view information in a BP. This implies different visibility of a BP from different partners, often based on the need-to-know principle. The following paragraphs report current research progress and discuss which aspects of BC can be useful in the allocation.

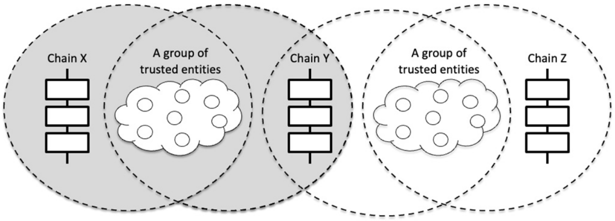


FIGURE 4. A notary scheme with transitive property.

Blockchain Studio [98] is an integrative work under Caterpillar. Authors perform improvement of Caterpillar by featuring it with a resource allocation framework based on Role-based Access Control (RBAC). Similar approach by translating BPMN into smart contracts, the framework empowers Caterpillar complier to automatically translate access control policies into smart contracts. Role and account assignments are done during BP modeling. This allows designers to incorporate and verify assignment activities. BPs are modeled by BPMN, and policies are expressed by XML. The framework is self-contained, presenting a complete set of artifacts with design, implementation, and evaluation. However, as identified by [127], the concerns from the organizational perspective is missing, meaning applying such approach to cross-organizational BPs may not be ready. The work needs an additional set of smart contracts to operate, resulting in the increase in gas. Other limitations are identified in [53], mostly related to the issues during runtime (L3). For instance, no security mechanisms are in place to continuously verify the status of smart contracts. It is rigid to accommodate changes as no synchronization of role and account are performed.

To relieve the weakness, Lopez-Pintado et al. [81] illustrated that a Dynamic Role Binding approach is capable of coping such changes during runtime (L4). This work extends Caterpillar with the control of actor-task relationships for each BP instance. As noted, PSL dictates the conditions to govern changes, for instance, setting a voting mechanism to reach agreement from pre-selected partners who are

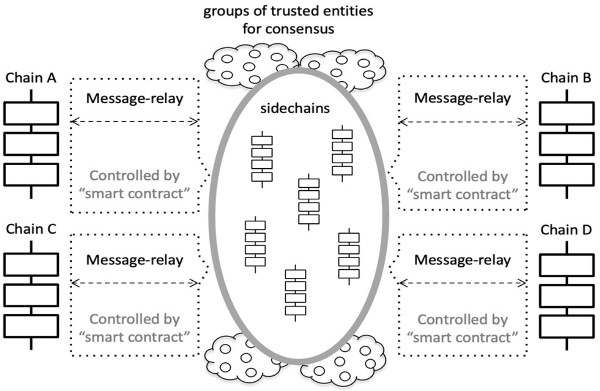


FIGURE 5. Blockchain interoperation with sidechain.

accounted for specific changes. Nomination net is introduced based on Petri Net to verify policy consistency. Automatic translation of PSL into smart contracts decouples policy specification activities from BC. This work is prototyped in Ethereum which encounters the problems from public BC and gas cost.

In this regard, one important issue related to decentralized BPM is the enforcement of local access control policies from distributed partners. Previous works assume these policies have been settled and partners agree on pre-defined policies and specifications, which will be translated into smart contracts. Later, they will be enforced globally. However, inditing these policies in compliance with disparate local policies is complicated. Conflicting policies need to be resolved. Using typical evaluation techniques will result in high overhead. To alleviate this, only one study is found by Akhtar et al. [5] which try to optimize the collection of individual policies towards composite unified policies. The policy specification follows XACML standard, which will be translated into smart contracts for enforcement and audit purpose. An incentive model based on game theoretic approach is discussed towards the possibility of auditing tasks, therefore preventing frauds from rational partners. However, this policy translation suffers from flexibility to address changes during runtime.

Taking the organizational perspective covering roles, accounts, and resources into consideration during configuration (L2), Sturm et al. [127] advanced their own work [128] pertaining to BC-based BP execution framework with resource allocation engine. Its capabilities are demonstrated through smart contracts that implement creation patterns, one of resource patterns found in [63], encompassing direct allocation, role-based allocation, and separation of duties. It expresses resource assignment constraints during configuration (L2), focusing on human collaborative tasks. However, it lacks implementation details regarding the process engine. Like many others, process models are created by BPMN, and translated into smart contracts. At this moment, the logics for resources allocations are strictly implemented by smart contracts. The work is comprehensively evaluated compared to other related approaches pertaining to BC-based BP executions, covering the works from Sturm et al. [128], Weber et al. [152], Blockchain Studio [98], and Madsen et al. [88]. Analysis is given, showing that the capability of the work is equivalent to Blockchain Studio [98] in terms of trust, and resource-awareness, with slight superiority on tamper protection. The only difference is the logics of specifying resource allocations. Sturm et al.'s approach relies on manually writing the logics directly to smart contracts, while Blockchain Studio [98] employs XML as a vehicle. The authors gave as important remark that permissioned BC is the most suitable type to restrict access to smart contracts. This proposal focuses on human collaborative tasks rather than non-human tasks, where processes can be made automated among IoT devices, external systems, etc. [53].

In summary, while many research attempts emphasize on BP executions on BC, only few have been found on resource assignments. This area is relatively new as the first document was published in 2019. Since then, following works utilize similar mechanisms by employing smart contracts as a building block to provide tamper-proof for resource assignment processes. We observe that the work by Akhtar et al. [5] can be integrated with Blockchain Studio [98] and Sturm et al. [127]. By taking local access control policies into account during configuration (L2), the work by Lopez-Pintado et al. [81] addresses dynamic role assignments at runtime (L3). However, extra smart contracts imply additional cost.

I. INTERPRETATION OF BP MODELS INTO BC

Several challenges towards translating standard notations into smart contract definition language have been studied recently. BPMN is a widely accepted standard for BP modeling, but its notations and semantics are not originally designed to work with BC. Still, BPMN is widely recognized as the main stream for modeling inter-organizational BPs [152]. Obviously, its insufficiency in expressiveness is a major hinderance to implement BPs on BC. This section evaluates prominent works that try to close this gap.

Ladleif et al. [77] suggest to refine and extend syntax and semantics of BPMN choreography diagrams to address BC. The idea is to modify choreography to be used as a decentralized approach for collaborations without a central entity to regulate business workflows. The main contribution is the extension and modification of components in existing diagram components to enhance expressiveness. The work focuses on inherent ownership and observability issues. Choreography is purely encoded by smart contracts. Directly modeling ownership, defining who are authorized for task executions and observation, by smart contracts programing can eliminate confusion and ambiguity of ownership assignments. Besides, BC is used as a complete and trusted logging mechanism to be observed by permitted partners. This proposal is backward compatible to BPMN. Authors describe operational semantics, and practicality is witnessed by testing through case studies using architecture presented in their previous work [152]. However, security issues such as confidentiality and privacy are left for future studies.

Due to the characteristics of consensus protocols used in public BC like PoW, Abid et al. [4] aim to tackle uncertainty of transaction confirmation (CH3). This can vary from a few seconds to several minutes or even hours. In many incidents, this uncertainty leads to serious violations or undesired outcomes towards temporal agreement, such as causing delay in payment or product delivery, and resulting in financial penalty. Unfortunately, BC does not have mechanisms to deal with temporal constraints. This work, rather than solving uncertainty of delay stemmed from consensus protocols, suggests a workaround technique using timestamp marking in smart contract code. There are two categories of this marking, relative and absolute temporal constraints on BPs. The notation of token from BPMN, which travels along a sequence flow, is modified to include timestamp as a marking variable, which will be used later for temporal restrictions. This information is transparent to partners. The work is integrated in Caterpillar for evaluation through case studies. However, quantitative experiments, where performance and scalability are analyzed, are not included. Moreover, the uncertainty of transaction conformation (CH3) originated from consensus protocols remains unsolved.

A framework by Bore et al. [24] focuses on usability by facilitating the processes of creating, updating, and using BP workflow on BC. Automatic translation from business models which are designed by a developed GUI to BC is proposed. The main advantage is BC technical knowledge is not much required. The framework is prototyped in Hyperledger Fabric. However, it is platform specific, while integrating to other BCs is not possible. The proposal is overviewed conceptually. Authors plan to implement the framework for evaluation and performance analysis.

In brief, a few attempts have been reported on the interpretation of BPs modelled by standard notations into smart contracts. Existing works enhance BPMN notations to facilitate BP development for different types of users [120]. However, the challenges related to BC are not completely addressed.

J. BC SUPPORTING QOS FOR SERVICE SELECTION IN BPS

The rapid growth of IoT and BC have enabled great opportunities to revolutionize BPM. IoT and BC are similar in decentralized topology, but different in philosophy. While IoT aims at functionalities, BC creates trust in an untrusted environment. Their mutual benefits will create the proliferation of BC-based IoT services. Therefore, BPM tends to face dynamicity and heterogeneity in using such services, which increasingly become part of BPs. While trust can be conceived when services are backed by BC, nonfunctional requirements like QoS are still the major decisive factors for service selection [54], [135]. In this circumstance, QoS, e.g., response time, availability, error rate, reputation, etc., will be the major determinants in service selection strategies. It provides insight information for business to select the best suited ones for their BPs. QoS is a widely-accepted approach to qualify services in open environments [139]. In BPM lifecycle, service selection activities first situate in configuration (L2) where services are selected and assigned for tasks. During runtime, service replacement triggers such activities (L4). However, one major obstacle is the trustiness of QoS information. Previous works often make an assumption that the information is innately trusted, while others utilize trusted third parties to endorse the information [145]. This scheme is facing scalability challenges and trustworthiness of the third parties becomes primary. Research in this area is relatively limited.

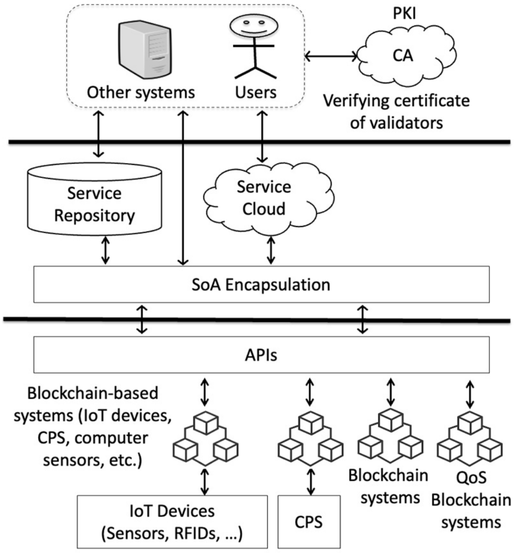

Other systems
Users
PKI
CA
Verifying certificate of validators
Service Repository
Service Cloud
SoA Encapsulation
APIs
Blockchain-based systems (IoT devices, CPS, computer sensors, etc.)
Blockchain systems
QoS Blockchain systems
IoT Devices (Sensors, RFIDs, ...)
CPS

FIGURE 6. An architecture of SoA-integrated BC as a service [147].

In this area, a series of publications are provided. A BC-based QoS scheme was firstly suggested in [145] to promote trust of QoS information. The framework illustrates a feedback scheme with an assumption that a service consumer is responsible for providing feedbacks, which will be calculated for QoS of a service. They propose a protocol programmed in smart contracts to enable trust to the entire processes including monitoring, collecting, and measuring QoS values. However, the cited work is conceptually presented. Scalability is a problem when applying to the real-world scenarios. Based on this work, a QoS scheme [143] is introduced to solve the limitations. Authors believe that to trust QoS information the processes of measuring and gathering QoS values, and partners that provide the values must be trusted. Scalability is mitigated by a layer architecture using zones and hubs along with the concept of transitive trust. This concept is exemplified in a large-scale IoT environment. The collection of QoS is done by agents in zones. Trust of agents is managed by a central hub using smart contracts to verify agents for trust. Specifically, smart contracts are used to specify the logics of registration of agents. Sharding technique is incorporated for faster rate of information processing. This scheme is implemented on permissioned BC running PBFT as a consensus protocol. Additionally, authors extended their own works to cover common specification patterns [141] found in Service-Based Application (SBAs) [22], which are used to evaluate QoS in industries. QoS related processes are implemented by smart contracts with special data structure to realize the patterns. However, the work series are facing a challenge of storage capacity. Besides, the processes of agent registration are assumed to be effective while, in the IoT context, it is hardly feasible to validate all device identities using the same rules implemented by smart contracts. More applications in the real-world scenarios needs to be conducted.

In summary, several BC-based systems are approached as services that provide supports to decentralized BPM. Exiting literature indicate how BC can assist during configuration, by relying on QoS information for service selection. Efficient QoS management and trust of the information can be achieved by BC. BC is used to record transactions and provides access to QoS. Smart contracts control the processes of QoS gathering and authenticate entities to activate smart contract functions. The QoS scheme assists BPs in service selection during configuration (L2) and on-the-fly service replacement during diagnosis and evaluation (L4). However, no research has been found regarding the integration of QoS scheme as a component in BPM. It is important to investigate a unified conceptual framework that integrates BC at each phase of BPM lifecycle towards a complete solution of BC applications in BPM.

VI. DISCUSSIONS AND RESEARCH DIRECTIONS

Successful applications of BC in BPM require business stakeholders to understand intrinsic values, challenges, and the impacts of BC on BPs. The technical challenges of BC when it was introduced in BPM have been discussed in [102]. Therefore, this section focuses only on the challenges of BC where satisfactory solutions have not been developed in the most recent development of BC in BPM.

A. INTEROPERATION OF BPS (CH1)

Interoperation of BPs requires compatibility of their respective IT systems. However, incompatibility of heterogeneous systems has been a long-standing hurdle for BPs across organizations. There is an emerging need for BPM to support interoperations of BPs in BPM [61]. Unluckily, adopting BC in BPs may increase the degree of heterogeneity of application platforms [117] and introduces grater challenges. Several attempts [17], [28] were made to support the direct interaction between BCs at the protocol level; however, it is expensive and hard to scale. Others adopted SoA as a key technology to smooth the interoperations [142], [147], [155]. SoA infrastructure has been well established; however, the integrations of services over SoA are not straightforward. Encapsulations of services affect the standardizations of APIs for external requests; since BC elements including type, consensus protocols, and delay of transaction confirmation (CH3), cause variants of interfaces and conditions. This becomes a significant challenge when incorporating with a public, permissioned, or private BCs. A possible solution is modular design by developing a middleware between the layers of SoA and BC to serve as an adapter to connect different BC modules [44], [132]. A middleware consists of a programming library with smart contracts that can be selected and composed as adapters during modeling processes [45]. The middleware approach is extensible to support interactions of different BCs using SoA in BPM.

Smart contracts can be embedded with voting mechanisms [81], [147] to take over consensus operations to solve the delay issue. This type of smart contracts is flexible to accommodate multiple techniques as well as changes occurring in runtime. Smart contracts for consensus can be viewed as a high-level consensus where changes are made by selected nodes.

At the business level, QoS, security, policies, and laws and regulations are defined and tailored for specified BPs and SLA constitutes the assertions of contractual clauses among partners. The requirements of interoperations defined in SLA are used to govern entire BP executions. Specification languages [133], [140], [148] as well as reasoning algorithms [146], [149] are essential to automate compliance checking processes in SLA. QoS schemes [143], [145] may also be exploited as trusted sources of QoS information which can be part of BC oracle. It has been discussed that traditional BPM involves a number of security issues such as data integrity and confidentiality, confidentiality of processes, trust of process executions, and data provenance [29], and it is necessary to integrate BC together with other technologies such as IoT, specification languages, SoA, Cloud computing, PKI, as a complete solution to BPM.

B. TRUST OF BC-BASED SYSTEMS (CH2)

BC promotes trust but it can introduce another dimension of threats. Therefore, BC cannot promote trust unless it is constructed correctly and can be verified by users. Our survey [105] showed that BC-enabled trust patterns can be synthesized as (1) trusted storages for data sharing such as using BC to document SLA, (2) transparent event logs to be traced by partners and to achieve the non-repudiation property, (3) smart contracts to guarantee correct logics in the executions of BPs, and (4) trust mechanisms to support BPs such as QoS and providing data exploitation schemes. It seems that BC is notable for trust in executing BPs; however, can we trust the services or entities that host BC systems without any doubt?

For a BC-enabled transaction, trust is not assured until it is confirmed. It means that the identifications of authorized validators and consensus protocols reflect trust of BPs. A permissioned BC is considered the best in reducing the latency of transaction confirmation (CH3). In addition, strategies in evaluating trust of BC applications are important. Using PKIs seems an ideal strategy in establishing initial trust [142]; however, adding exclusive attributes to endorse BC applications in PKI certificate requires other strategies such as proof of implementation [147]. Particularly, trust depends greatly on the selections of validators who execute the consensus operations. It becomes possible that trust cannot be redeemed by partners if collusion of validators can falsify the previously validated transactions. Therefore, it is essential for users to endorse the proofs of validators by PKI certificates and using formal specification languages to specify validator attributes can automate compliance checking processes. From this perspective, the methodologies from *design science* and *action research* can be applied to identify the attributes to be endorsed in validations [62].

C. DELAY OF TRANSACTION CONFIRMATION (CH3)

Consensus algorithms determine the time required to confirm transactions. Algorithms in permissioned or private BCs usually allow fast and definite confirmation, while algorithms in public BCs confirm transactions with probability. For example, a Bitcoin PoW suggests that transactions are confirmed when it is located in the depth of six blocks and the confirmation time varies from minutes to hours [155]. Confirming transactions with excessive and inconsistent delay is the main cause that must be well aware when incorporating public BCs in time-sensitive BPs [136], [147].

It is unavoidable to wait some time until consensus algorithms confirm transactions, and confirmation delay is natural to any consensus algorithm even ones used in permissioned or private BCs. A few researchers specifically discussed this issue [44], [45], [142]. Confirmation delays in BC interactions are simply marked when modelling BPs [45]; while others addressed the issue by observing maximum delay in permissioned or private BCs [142]. In most cases, permissioned BCs permit quicker and cheaper transaction validation due to the limited number of validators.

D. IRREVERSIBILITY (CH4)

BPs have to deal with changes during execution, but this is contradicted with BC immutability. A BP may involve changes regularly such as the changes in organizational policies, laws and regulations, or technical upgrades in its execution infrastructure. Note that tasks have been completed and confirmed are irreversible [20], [93] (CH4). *Reversibility* refers to reverse unwanted or faulty executions for the changes made to a BP, and irreversibility is sometimes coupled with a downside coupling with complexity and cost. Irreversibility affects trust of BC. It requires a new fork from a main chain, and some confirmed transactions become invalidated. Making BC transactions reversible diminishes trust that BC provides.

From existing literature, BP changes can be done in two ways: (1) generating outputs by anticipating all possible changes to support *undo* operations [43] and (2) triggering changes by adding high-level consensus using smart contracts [82]. For example, a voting mechanism can be implemented by smart contracts as an upper layer consensus to commit changes; this offers flexibility in the changes of partners and their associated tasks [73], [155]. Note that substantial overheads on resource consumptions are accounted when smart contracts are implemented to cope with changes.

E. SMART CONTRACTS FOR BPS

Smart contracts are widely applied to BPs to define constraints of businesses and coordinate BP executions. Smart contracts act as a central mediator to manage invocation and monitor executions in a decentralized manner. In some settings, smart contracts are associated with a serious behavior known as *state distinction*. When some types of consensus protocols such as PoW are applied, distinction of any two states may cause undesired executions [43]. For example, two activities A and B come in order, i.e., B must not be executed until the conditions for A are confirmed; otherwise, B involves in a risk of being invalidated or reversed if the conditions of A are not accepted in a consensus chain. When a BP faces two different states, and only confirmed states can be proceeded safely. In such case, how to deal with confirmation delay becomes a major concern; addressing the issue may cause resource wastes and lessen performance of BPM [23]. No satisfactory solution has been found when this paper is written.

F. BUSINESS PROCESS COMPLIANCE (BPC)

There are several concerns in adopting BC technology in organizations, actually in every new technology adoption. BPC was to assure important aspects for digital transformation of current BPs as well as involved stakeholders. BPC was essential to successful technology adoption. Five challenges in performing BPC are discussed below.

Technical challenges indicate technical requirements that will be interpreted into artifacts. This process is time consuming and costly. The difference of information systems within organizations and from interacting business partners often cause incompatibility that hinders the success of new technical solutions.

Organizational challenges regard the variation of partners that had unique organization structures, technologies, scales, security standards, and legislation, but required to interact with each other. BC can be used as standard for BP interoperation. However, additional agreements and BP reengineering tailored to BC must be considered in defining new BPs.

Legal challenges have recently become a big issue that regulate parties to adopt BC technology. We are facing many requirements of policies, laws, and regulations with uncertainty and ambiguity. Organizations must be well-aware of these requirements as well as anticipate future changes. It is a long-term process in transforming BPs with BC which require updates regularly.

Human-centered challenges are the most important and the most difficult elements to deal with changes, especially associated with new processes and technologies. The unreadiness of human factors such as knowledge, current workloads, and resistant to changes, become major obstacles.

Economic challenges relate to Return-of-Investment in adopting BC technology and to reengineer BPs. Adopting BC needs long-term investment to new information systems

TABLE 3. Summary of research articles.

Articles	Area of Applicability															
	Challenges				BPM characteristics								BPM lifecycle			
	CH1	CH2	CH3	CH4	C1	C2	C3	C4	C5	C6	C7	C8	L1	L2	L3	L4
A. Blockchain-assisted BPM																
Cen et al. [30] Fang et al. [47] Chang et al. [32]		✓				✓		✓	✓			✓	✓			
Alves et al. [9]		✓				✓		✓	✓			✓	✓	✓		
Panduwinata and Yugopuspito [108]		✓				✓		✓	✓		✓		✓			
Scientific workflow																
ProChain [33] SciBlock [49]		✓			✓			✓	✓			✓	✓			
Karastoyanova and Stage [69]		✓			✓		✓		✓			✓	✓			
B. Blockchain as an Infrastructure																
Process-centric																
García-Bañuelos et al. [52]		✓				✓		✓		✓					✓	
Weber et al. [152]		✓				✓		✓		✓		✓	✓	✓	✓	
Sturm et al. [128]		✓				✓		✓	✓			✓	✓	✓	✓	
Nakamura et al. [106]		✓				✓		✓		✓		✓		✓	✓	
Lorikeet [130]		✓				✓		✓		✓				✓	✓	
Caterpillar [84] [85] Madsen et al. [88]		✓				✓		✓		✓		✓			✓	
Data-centric																
Hull et al. [67,68]		✓				✓	✓									
B-MERODE [11]		✓				✓	✓		✓	✓		✓		✓		
Smart contract-less																
Härer [58]		✓				✓		✓	✓	✓		✓	✓	✓		
Evermann and Kim [43]		✓				✓		✓		✓		✓	✓		✓	
C. Flexibility																
L´opez-Pintado et al. [81]		✓				✓	✓			✓		✓			✓	
Lopez-Pintado et al. [83]		✓		✓	✓	✓	✓			✓		✓			✓	✓
Klinger and Bodendorf [72]		✓		✓	✓	✓	✓		✓	✓		✓			✓	✓
Lopez-Pintado et al. [82]		✓			✓		✓			✓		✓			✓	✓
Evermann and Kim [43] Klinger et al. [73]		✓		✓		✓	✓		✓	✓	✓		✓	✓	✓	
Madsen et al. [88]		✓				✓		✓		✓		✓				✓
D. Runtime Monitoring and Traceability																
Weber et al. [152] Prybila et al. [109] Haarmann et al. [57] Di Ciccio et al. [35] Silva et al. [124]		✓														✓
Meroni and Plebani [99] , Meroni et al. [100]		✓								✓		✓				✓
E. Process Mining on Blockchain																
GraphSense [60]																✓
Duchmann and Koschmider [41]		✓								✓					✓	✓
Klinkmuller et al. [74] Muhlberger et al. [104]		✓								✓		✓			✓	✓
F. Smart Contract Verification																
Amani et al. [10]		✓				✓		✓	✓					✓		
ContractLarva [15] Ellul and Pace [42] and Yu et al. [156] Azzopardi et al. [16]		✓					✓			✓					✓	
Duchmann and Koschmider [41]																
FsolidM framework [91]		✓						✓	✓				✓			
Hu et al. [64]						✓		✓	✓				✓			
G. Blockchain Interoperability																
Notary scheme [28] sidechain [18]		✓						✓	✓	✓		✓	✓	✓	✓	
BlockME [44]	✓	✓	✓			✓		✓			✓			✓	✓	
BlockME2 [45]	✓	✓	✓			✓		✓	✓		✓		✓	✓	✓	
Viriyasitavat et al. [142]	✓	✓	✓			✓		✓	✓			✓	✓			
Viriyasitavat et al. [147]	✓	✓	✓			✓		✓	✓			✓	✓	✓		
Xu and Viriyasitavat [155]		✓	✓			✓		✓	✓			✓	✓	✓		
H. Resource-Aware Business Process																
Blockchain Studio [98]		✓				✓		✓	✓			✓		✓		
Lopez-Pintado et al. [81]		✓				✓	✓			✓		✓			✓	
Akhtar et al. [5] Sturm et al. [127]		✓				✓		✓	✓			✓		✓		
I. Interpretation of Business Process Models																
Ladleif et al. [77]		✓				✓		✓	✓			✓		✓		✓
Abid et al. [4]			✓			✓		✓	✓			✓		✓	✓	
Bore et al. [24]		✓				✓		✓	✓				✓			
J. Blockchain Support QoS for BPs																
Viriyasitavat et al. [23] Viriyasitavat et al. [141]		✓							✓					✓		
Viriyasitavat et al. [143]		✓	✓						✓					✓		

to define, manage and execute BPs. Additionally, legacy systems must be updated to work with BC-enabled BPs. Cost-benefit models are required to evaluate the use of BC.

G. APPLICATIONS IN BPS

Lastly, numerous BC applications have been developed for BPs, and the systems are tailored to specific characteristics with different strengths and weaknesses. Users are facing the challenge in selecting the most suitable BC to meet their needs. The comparisons of popular platforms were made to guide users to make their choices [87]; however, the appropriate assessment of a certain BC depends on many factors varying in different contexts, such as business types, enterprise resources, network capacities, and maturity of business environments [40].

An assessment model plays an important role to help select right BCs for specific applications [76], [154]. Characteristics and business variables of BPs must be taken into account [21]. Multi-Criteria Decision Making (MCDM) can be an effective approach to model the dependence of BC performances on BPs [48], [129]. It is desirable to automate assessment processes according to specified requirements and architectural patterns in different businesses [125]. Trust is considered a primary factor in selecting BCs [123], and the action design and situational methods can be employed to develop some value-added BC systems [51].

Existing assessment models were developed to evaluate general information systems. Even though parts of evaluation metrics are applied to BPM, it involves special characteristics related to business operations such as interoperation requirements, security assurance, compatibility, and time-sensitiveness. Taking an example of latency in a BP [56], the assessment model must take into account the types and configurations of BC to balance the trade-off between trust adequacy and operational viability. Due to inexplicit influence of these factors on latency, evaluation of BCs may be benefited from model-driven engineering approaches [132]. They have been used to accelerate the adoption of new technologies. However, prerequisite is a good understanding of BC-specific configurations that are exclusive from traditional systems in terms of properties and behaviors [45].

BPM at the organization level should be capable of engaging people and BPs together with newly adopted technologies. Many researchers suggest to use Ethereum for BPM due to its advancement, high popularity, and reliability in comparison with other smart contract-enabled BC platforms [93]. Adopting Ethereum increases additional cost in executing BPs. The cost analysis from monetary and operation aspects [114], [115] suggests that the cost of BC-based BPM is almost tripled in comparison with conventional solutions with cloud technology or traditional database. To fully explore BC potentials, existing BC applications should be refined based on best practices to reduce running cost [101].

BC-enabled BPM should automate the processes of creating, updating and executing BPs regardless of BC types [24] equipped with agnostic smart contracts. However, In developing a BC-enabled BPM, developers should be well educated to understand the impacts of BCs on BPs, especially the latency of transaction confirmation (CH3) caused by consensus algorithms [43]. BC is not the only technological option for BPM, the decision of adopting BCs must be aligned with business strategies. In many cases, BCs have to be integrated with cutting-edge technologies, such as machine learning, IoT, big data analytics, Cloud and edge computing, artificial intelligence, and robotic process automation to advance BPM [97]. Numerous opportunities are emerging in applying BCs to BPM.

VII. SUMMARY

Since its emergence, BC technology has been evolved continuously and has resulted in many variations of BC types. BC technology has brought numerous opportunities in innovating the management of BPs [40]: firstly, BC can be used as a recoding mechanism to improve business trust. BC warrantees the immutability of BP executions to reduce any dispute that is usually costly in current business practices. Moreover, BC has been envisioned to obtain trust in an untrusted environment; it is used to establish trusted infrastructure to execute BPs. Note that the trust has been a long-standing obstacle for effective collaborations across businesses, especially in dynamic and distributed environments.

The presented work was motivated by our observation that no comprehensive surveys had been published on the applications of BC in BPM, and we aimed to gain the state-of-the-art of the research in this field and help practitioners to understand the maturity, benefits, and challenges of BC potentials in BPM. It was found that the study of adopting BC in BPM had attracted a great deal of attention that was evidenced by a rapidly growing number of relevant articles on the integration of BC technology in BPM. The paradigms of IoT-enabled BPM were shifted from persistent to transient, from static to dynamic, and from centralized to decentralized; while BC had been intensively studied as a promising solution to assure the trustiness for both of business processes and their executions in decentralized BPM, future research efforts were needed to meet the challenges involved in interoperation, determination of trusted entities, time-sensitive confirmation of execution, and irreversibility. It was worth to note that this survey focused on the challenges on methodological frameworks and technologies of adopting BC in business lifecycles from the application perspective, the challenges on infrastructure, standards, models, methods, and algorithms have yet been discussed thoroughly.

REFERENCES

[1] W. M. P. van der Aalst, M. Pesic, and H. Schonenberg, "Declarative Workflows: Balancing between flexibility and support," *Comput. Sci.-Res. Develop.*, vol. 23, no. 2, pp. 99–113, May 2009, doi: 10.1007/s00450-009-0057-9.

[2] W. M. P. van der Aalst, "Business process management: A comprehensive survey," *ISRN Softw. Eng.*, vol. 2013, pp. 1–37, Feb. 2013, doi: 10.1155/2013/507984.

[3] M. P. Wil Van Der Aalst, "Business process management demystified: A tutorial on models," in *Lectures on Concurrency and Petri Nets* (Lecture Notes in Computer Science), vol. 3098, 2004, pp. 1–65, doi: 10.1007/978-3-540-27755-2_1.

[4] A. Abid, S. Cheikhrouhou, and M. Jmaiel, "Modelling and executing time-aware processes in trustless blockchain environment," in *Proc. Int. Conf. Risks Secur. Internet Syst.* (Lecture Notes in Computer Science). Cham, Switzerland: Springer, 2020, pp. 325–341, doi: 10.1007/978-3-030-41568-6_21.

[5] A. Akhtar, B. Shafiq, J. Vaidya, A. Afzal, S. Shamail, and O. Rana, ''Blockchain based auditable access control for distributed business processes,'' in *Proc. IEEE 40th Int. Conf. Distrib. Comput. Syst. (ICDCS)*, Nov. 2020, pp. 212–222, doi: 10.1109/ICDCS47774.2020.00015.

[6] F. A. Alaba, M. Othman, I. A. T. Hashem, and F. Alotaibi, ''Internet of Things security: A survey,'' *J. Netw. Comput. Appl.*, vol. 88, pp. 10–28, Jun. 2017, doi: 10.1016/j.jnca.2017.04.002.

[7] M. S. Ali, M. Vecchio, M. Pincheira, K. Dolui, F. Antonelli, and M. H. Rehmani, ''Applications of blockchains in the Internet of Things: A comprehensive survey,'' *IEEE Commun. Surveys Tuts.*, vol. 21, no. 2, pp. 1676–1717, 2nd Quart., 2019, doi: 10.1109/COMST.2018.2886932.

[8] M. Almakhour, L. Sliman, A. E. Samhat, and A. Mellouk, ''On the verification of smart contracts: A systematic review,'' in *Proc. Int. Conf. Blockchain* (Lecture Notes in Computer Science), vol. 12404, Sep. 2020, pp. 94–107, doi: 10.1007/978-3-030-59638-5_7.

[9] P. Alves, R. Paskin, I. Frajhof, Y. Miranda, J. Jardim, J. Cardoso, E. Tress, R. Ferreira da Cunha, R. Nasser, and G. Robichez, ''Exploring blockchain technology to improve multi-party relationship in business process management systems,'' in *Proc. 22nd Int. Conf. Enterprise Inf. Syst.*, Dec. 2020, pp. 817–825, doi: 10.5220/0009565108170825.

[10] S. Amani, M. Bégel, M. Bortin, and M. Staples, ''Towards verifying ethereum smart contract bytecode in Isabelle/HOL,'' in *Proc. 7th ACM SIGPLAN Int. Conf. Certified Programs Proofs*, Jan. 2018, pp. 66–77, doi: 10.1145/3167084.

[11] V. A. de Sousa, C. Burnay, and M. Snoeck, ''B-MERODE: A model-driven engineering and artifact-centric approach to generate blockchain-based information systems,'' in *Proc. Int. Conf. Adv. Inf. Syst. Eng.* (Lecture Notes in Computer Science). Cham, Switzerland: Springer, 2020, pp. 117–133, doi: 10.1007/978-3-030-49435-3_8.

[12] H. Ariouat, C. Hanachi, E. Andonoff, and F. Benaben, ''A Conceptual Framework for Social Bus. Process Management,'' *Proc. Comput. Sci.*, vol. 112, pp. 703–712, Apr. 2017, doi: 10.1016/j.procs.2017.08.151.

[13] N. Atzei, M. Bartoletti, and T. Cimoli, ''A survey of attacks on ethereum smart contracts (SoK),'' in *Proc. Int. Conf. Princ. Secur. Trust* (Lecture Notes in Computer Science), vol. 10204, 2017, pp. 164–186, doi: 10.1007/978-3-662-54455-6_8.

[14] C. Ayora, V. Torres, B. Weber, M. Reichert, and V. Pelechano, ''VIVACE: A framework for the systematic evaluation of variability support in process-aware information systems,'' *Inf. Softw. Technol.*, VOL. 57, pp. 248–276, Jan. 2015, doi: 10.1016/j.infsof.2014.05.009.

[15] S. Azzopardi, J. Ellul, and J. G. Pace, ''Monitoring smart contracts: Contractlarva and open challenges beyond,'' in *Proc. Int. Conf. Runtime Verification* (Lecture Notes in Computer Science). New York, NY, USA: Springer Verlag, 2018, pp. 113–137, doi: 10.1007/978-3-030-03769-7_8.

[16] S. Azzopardi, G. J. Pace, and F. Schapachnik, ''On observing contracts: Deontic contracts meet smart contracts,'' in *Legal Knowledge and Information Systems*. Amsterdam, The Netherlands: IOS Press, 2018, pp. 21–30, doi: 10.3233/978-1-61499-935-5-21.

[17] A. Back, M. Corallo, L. Dashjr, M. Friedenbach, G. Maxwell, A. Miller, A. Poelstra, J. Timón, and P. Wuille. (2014). *Enabling Blockchain Innovations with Pegged Sidechains*. Accessed: Apr. 19, 2021. [Online]. Available: http://kevinriggen.com/files/sidechains.pdf

[18] S. A. Back, M. Corallo, L. Dashjr, M. Friedenbach, G. Maxwell, K. Andrew Miller, A. Poelstra, and N. J. Tim. (2014). *Enabling Blockchain Innovations with Pegged*. Accessed: Apr. 22, 2019. [Online]. Available: https://www.semanticscholar.org/paper/Enabling-Blockchain-Innovations-with-Pegged-Back-Corallo/1b23cd2050d5000c05e1da3c9997b308ad5b7903

[19] A. Baouab, O. Perrin, and C. Godart, ''An event-driven approach for runtime verification of inter-organizational choreographies,'' in *Proc. IEEE Int. Conf. Services Comput.*, Jul. 2011, pp. 640–647, doi: 10.1109/SCC.2011.55.

[20] J. Becker, P. Delfmann, M. Eggert, and S. Schwittay, ''Generalizability and applicability of model-based business process compliance-checking approaches—A state-of-the-art analysis and research roadmap,'' *Bus. Res.*, vol. 5, no. 2, pp. 221–247, Nov. 2012, doi: 10.1007/BF03342739.

[21] M. Belotti, N. Bozic, G. Pujolle, and S. Secci, ''A vademecum on blockchain technologies: When, which, and how,'' *IEEE Commun. Surveys Tuts.*, vol. 21, no. 4, pp. 3796–3838, 2019, doi: 10.1109/COMST.2019.2928178.

[22] D. Bianculli, C. Ghezzi, C. Pautasso, and P. Senti, ''Specification patterns from research to industry: A case study in service-based applications,'' in *Proc. 34th Int. Conf. Softw. Eng. (ICSE)*, Jun. 2012, pp. 968–976, doi: 10.1109/ICSE.2012.6227125.

[23] S. Biswas, K. Sharif, F. Li, S. Maharjan, S. P. Mohanty, and Y. Wang, ''PoBT: A lightweight consensus algorithm for scalable IoT business blockchain,'' *IEEE Internet Things J.*, vol. 7, no. 3, pp. 2343–2355, Mar. 2020, doi: 10.1109/JIOT.2019.2958077.

[24] N. Bore, A. Kinai, J. Mutahi, D. Kaguma, F. Otieno, S. L. Remy, and K. Weldemariam, ''On using blockchain based workflows,'' in *Proc. IEEE Int. Conf. Blockchain Cryptocurrency (ICBC)*, May 2019, pp. 112–116, doi: 10.1109/BLOC.2019.8751446.

[25] A. R. Botha, A. R. Botha, and H. P. J. Eloff, ''A security interpretation of the workflow reference model,'' in *Proc. Inf. Secur. (FROM SMALL)*, vol. 9, May 2021, pp. 43–51.

[26] R. Breu, S. Dustdar, J. Eder, C. Huemer, G. Kappel, J. Kopke, P. Langer, J. Mangler, J. Mendling, G. Neumann, S. Rinderle-Ma, S. Schulte, S. Sobernig, and B. Weber, ''Towards living inter-organizational processes,'' in *Proc. IEEE 15th Conf. Bus. Informat.*, Jul. 2013, pp. 363–366, doi: 10.1109/CBI.2013.59.

[27] A. Bridgwater. (May 13, 2021). *Blockchains are Verticalizing, So We Need Interoperability*. [Online]. Available: https://www.forbes.com/sites/adrianbridgwater/2018/02/07/blockchains-are-verticalizing-so-we-need-interoperability/?sh=66c722a77ab9

[28] V. Buterin. (2016). *Chain Interoperability*. Accessed: Apr. 19, 2021. [Online]. Available: http://andolfatto.blogspot.ca/2015/02/fedcoin-on-desirability-of-government.html

[29] B. Carminati, E. Ferrari, and C. Rondanini, ''Blockchain as a platform for secure inter-organizational business processes,'' in *Proc. IEEE 4th Int. Conf. Collaboration Internet Comput. (CIC)*, Oct. 2018, pp. 122–129, doi: 10.1109/CIC.2018.00027.

[30] Y. Cen, H. Wang, and X. Li, ''Improving business process interoperability by shared ledgers,'' in *Proc. 6th Int. Conf. Informat., Environ., Energy Appl.*, Mar. 2017, pp. 89–93, doi: 10.1145/3070617.3070631.

[31] S. E. Chang and Y. Chen, ''When blockchain meets supply chain: A systematic literature review on current development and potential applications,'' *IEEE Access*, vol. 8, pp. 62478–62494, 2020, doi: 10.1109/ACCESS.2020.2983601.

[32] S. E. Chang, Y.-C. Chen, and M.-F. Lu, ''Supply chain re-engineering using blockchain technology: A case of smart contract based tracking process,'' *Technol. Forecasting Social Change*, vol. 144, pp. 1–11, Jul. 2019, doi: 10.1016/j.techfore.2019.03.015.

[33] W. Chen, X. Liang, J. Li, H. Qin, Y. Mu, and J. Wang, ''Blockchain based provenance sharing of scientific workflows,'' in *Proc. IEEE Int. Conf. Big Data (Big Data)*, Dec. 2018, pp. 3814–3820, doi: 10.1109/BigData.2018.8622237.

[34] C. Di Ciccio, A. Cecconi, M. Dumas, L. García-Bañuelos, O. López-Pintado, Q. Lu, J. Mendling, A. Ponomarev, A. Binh Tran, and I. Weber, ''Blockchain support for collaborative business processes,'' *Informatik Spektrum*, vol. 42, no. 3, pp. 182–190, Jun. 2019, doi: 10.1007/S00287-019-01178-X.

[35] C. D. Ciccio, A. Cecconi, J. Mendling, D. Felix, D. Haas, D. Lilek, F. Riel, A. Rumpl, and P. Uhlig, ''Blockchain-based traceability of inter-organisational bus. processes,'' in *Proc. Int. Symp. Bus. Model. Softw. Design* (Lecture Notes in Business Information Processing). Cham, Switzerland: Springer Verlag, 2018, pp. 56–68, doi: 10.1007/978-3-319-94214-8_4.

[36] C. D. Ciccio, G. Meroni, and P. Plebani, ''Business process monitoring on blockchains: Potentials and challenges,'' in *Proc. Int. Conf. Bus. Process Model., Develop. Support* (Lecture Notes in Business Information Processing). Cham, Switzerland: Springer, 2020, pp. 36–51, doi: 10.1007/978-3-030-49418-6_3.

[37] N. D. Hacks. (2017). *Scams and Attacks: Blockchain's 2017 Disasters–CoinDeskm*. Accessed: May 14, 2021. [Online]. Available: https://www.coindesk.com/hacks-scams-attacks-blockchains-biggest-2017-disasters

[38] L. G. Jan Dietz, *Enterprise Ontology: Theory and Methodology*. Berlin, Germany: Springer, 2006, doi: 10.1007/3-540-33149-2.

[39] T. T. A. Dinh, R. Liu, M. Zhang, G. Chen, B. C. Ooi, and J. Wang, ''Untangling blockchain: A data processing view of blockchain systems,'' *IEEE Trans. Knowl. Data Eng.*, vol. 30, no. 7, pp. 1366–1385, Jul. 2018, doi: 10.1109/TKDE.2017.2781227.

[40] L. X. Downey, F. Bauchot, and J. Roling, ''Blockchain for business value: A contract and work flow management to reduce disputes pilot project,'' *IEEE Eng. Manag. Rev.*, vol. 46, no. 4, pp. 86–93, Dec. 2018, doi: 10.1109/EMR.2018.2883328.

[41] F. Duchmann and A. Koschmider, ''Validation of smart contracts using process mining,'' in *Proc. ZEUS, CEUR Workshop*, Bayreuth, Germany, 2019, pp. 13–16.

[42] J. Ellul and G. J. Pace, ''Runtime verification of ethereum smart contracts,'' in *Proc. 14th Eur. Dependable Comput. Conf. (EDCC)*, Sep. 2018, pp. 158–163, doi: 10.1109/EDCC.2018.00036.

[43] J. Evermann and H. Kim, ''Workflow management on the blockchain—Implications and recommendations,'' 2019, *arXiv:1904.01004*.

[44] G. Falazi, M. Hahn, U. Breitenbücher, and F. Leymann, ''Modeling and execution of blockchain-aware business processes,'' in *Software-Intensive Cyber-Physical Systems*. London, U.K.: Springer, 2019, pp. 105–116, doi: 10.1007/s00450-019-00399-5.

[45] G. Falazi, M. Hahn, U. Breitenbucher, F. Leymann, and V. Yussupov, ''Process-based composition of permissioned and permissionless blockchain smart contracts,'' in *Proc. IEEE 23rd Int. Enterprise Distrib. Object Comput. Conf. (EDOC)*, Oct. 2019, pp. 77–87, doi: 10.1109/EDOC.2019.00019.

[46] G. Falazi, M. Hahn, U. Breitenbücher, and F. Leymann, ''Transactional properties of permissioned blockchains,'' in *Software-Intensive Cyber-Physical Systems*. Cham, Switzerland: Springer, 2020, pp. 49–61, doi: 10.1007/s00450-019-00411-y.

[47] Y. Fang, X. Tang, M. Pan, and Y. Yu, ''A workflow interoperability approach based on blockchain,'' in *Proc. Int. Conf. Internet Vehicles* (Lecture Notes in Computer Science). Cham, Switzerland: Springer, 2020, pp. 303–317, doi: 10.1007/978-3-030-38651-1_24.

[48] S. Farshidi, S. Jansen, S. Espana, and J. Verkleij, ''Decision support for blockchain platform selection: Three industry case studies,'' *IEEE Trans. Eng. Manag.*, vol. 67, no. 4, pp. 1109–1128, Nov. 2020, doi: 10.1109/TEM.2019.2956897.

[49] D. Fernando, S. Kulshrestha, J. D. Herath, N. Mahadik, Y. Ma, C. Bai, P. Yang, G. Yan, and S. Lu, ''SciBlock: A blockchain-based tamper-proof non-repudiable storage for scientific workflow provenance,'' in *Proc. IEEE 5th Int. Conf. Collaboration Internet Comput. (CIC)*, Dec. 2019, pp. 81–90, doi: 10.1109/CIC48465.2019.00019.

[50] P. Franco, *Understanding Bitcoin*. Hoboken, NJ, USA: Wiley, doi: 10.1002/9781119019138.

[51] G. Fridgen, J. Lockl, S. Radszuwill, A. Rieger, A. Schweizer, and N. Urbach, ''A solution in search of a problem: A method for the development of blockchain use cases,'' in *Proc. AMCIS*, Aug. 2018, pp. 1256–1266. [Online]. Available: https://aisel.aisnet.org/amcis2018/StrategicIT/Presentations/14 2018

[52] L. García-Bañuelos, A. Ponomarev, M. Dumas, and I. Weber, ''Optimized execution of business processes on blockchain,'' 2016, *arXiv:1612.03152*.

[53] J. A. Garcia-Garcia, N. Sanchez-Gomez, D. Lizcano, M. J. Escalona, and T. Wojdynski, ''Using blockchain to improve collaborative business process management: Systematic literature review,'' *IEEE Access*, vol. 8, pp. 142312–142336, 2020, doi: 10.1109/ACCESS.2020.3013911.

[54] N. Gibelin and M. Makpangou, ''Efficient and transparent web-services selection,'' in *Proc. Int. Conf. Service-Oriented Comput.* (Lecture Notes in Computer Science). Berlin, Germany: Springer, 2005, pp. 527–532, doi: 10.1007/11596141_44.

[55] A. Gurtu and J. Johny, ''Potential of blockchain technology in supply chain management: A literature review,'' *Int. J. Phys. Distrib. Logistics Manag.*, vol. 49, no. 9, pp. 881–900, Nov. 2019, doi: 10.1108/IJPDLM-11-2018-0371.

[56] S. Haarmann, ''Estimating the duration of blockchain-based bus. processes using simulation,'' in *Proc. 11th Central Eur. Workshop Services Composition (ZEUS)*, 2005, pp. 24–31.

[57] S. Haarmann, K. Batoulis, A. Nikaj, and M. Weske, ''DMN decision execution on the ethereum blockchain,'' IN *Proc. Int. Conf. Adv. Inf. Syst. Eng.* (Lecture Notes in Computer Science). Berlin, Germany: Springer Verlag, 2018, pp. 327–341, doi: 10.1007/978-3-319-91563-0_20.

[58] H. Felixrer, ''Decentralized process modeling and instance tracking secured by a blockchain,'' *Proc. 26th Eur. Conf. Inf. Syst.*, 2018, pp. 1–5. [Online]. Available: https://fis.uni-bamberg.de/handle/uniba/44162

[59] D. Harz and W. Knottenbelt, ''Towards safer smart contracts: A survey of languages and verification methods,'' 2018, *arXiv:1809.09805*.

[60] B. Haslhofer, R. Karl, and E. Filtz, ''O Bitcoin where art thou? Insight into large-scale transaction graphs,'' in *Proc. SEMANTICS*, 2016, pp. 1356–1367.

[61] T. Henry, N. Laga, J. Hatin, W. Gaaloul, and I. Boughzala, ''Cross-collaboration processes based on blockchain and IoT: A survey,'' in *Proc. Annu. Hawaii Int. Conf. Syst. Sci.*, 2021, pp. 4291–4300, doi: 10.24251/hicss.2021.521.

[62] R. H. Von Alan, S. T. March, J. Park, and S. Ram, ''Design science in information systems research,'' *MIS Quart.*, vol. 28, no. 1, pp. 75–105, Mar. 2004, doi: 10.2307/25148625.

[63] A. H. M. T. Hofstede, ''Workflow patterns: On the expressive power of (Petri-net-based) workflow languages,'' *Daimi, Univ. Aarhus*, vol. 560, pp. 1–20, Oct. 2003.

[64] W. Hu, Z. Fan, and Y. Gao, ''Research on smart contract optimization method on blockchain,'' *IT Prof.*, vol. 21, no. 5, pp. 33–38, Sep./Oct. 2019, doi: 10.1109/MITP.2019.2923604.

[65] S. Huckle, R. Bhattacharya, M. White, and N. Beloff, ''Internet of Things, blockchain and shared economy applications,'' *Proc. Comput. Sci.*, vol. 98, pp. 461–466, Sep. 2016, doi: 10.1016/j.procs.2016.09.074.

[66] A. Hughes, A. Park, J. Kietzmann, and C. Archer-Brown, ''Beyond bitcoin: What blockchain and distributed ledger technologies mean for firms,'' *Bus. Horizons*, vol. 62, no. 3, pp. 273–281, 2019, doi: 10.1016/j.bushor.2019.01.002.

[67] R. Hull, ''Blockchain: Distributed event-based processing in a data-centric world: Extended abstract,'' in *Proc. 11th ACM Int. Conf. Distrib. Event-Based Syst.*, Jun. 2017, doi: 10.1145/3093742.3097982.

[68] R. Hull, V. S. Batra, Y. M. Chen, A. Deutsch, F. F. T. Heath, and V. Vianu, ''Towards a shared ledger business collaboration language based on data-aware processes,'' in *Proc. Int. Conf. Service-Oriented Comput.* (Lecture Notes in Computer Science). Cham, Switzerland: Springer Verlag, 2016, pp. 18–36, doi: 10.1007/978-3-319-46295-0_2.

[69] D. Karastoyanova and L. Stage, ''Towards collaborative and reproducible scientific experiments on blockchain,'' in *Proc. Int. Conf. Adv. Inf. Syst. Eng.* (Lecture Notes in Business Information Processing). Cham, Switzerland: Springer Verlag, 2018, pp. 144–149, doi: 10.1007/978-3-319-92898-2_12.

[70] R. Khalil and A. Gervais, ''Revive: Rebalancing off-blockchain payment networks,'' in *Proc. ACM SIGSAC Conf. Comput. Commun. Secur.*, ACM Press, New York, New York, USA Oct. 2017, pp. 439–453, doi: 10.1145/3133956.3134033.

[71] H. M. Kim and M. Laskowski, ''Toward an ontology-driven blockchain design for supply-chain provenance,'' *Intell. Syst. Accounting, Finance Manag.*, vol. 25, no. 1, pp. 18–27, Jan. 2018, doi: 10.1002/isaf.1424.

[72] P. Klinger and F. Bodendorf, ''Blockchain-based cross-organizational execution framework for dynamic integration of process collaborations,'' in *Proc. 15th Int. Conf. Bus. Inf. Syst.*, 2020, pp. 1–10, doi: 10.30844/wi_2020_i2.

[73] P. Klinger, L. Nguyen, and F. Bodendorf, ''Upgradeability concept for collaborative blockchain-based bus. process execution framework,'' in *Proc. Int. Conf. Blockchain*, vol. 12404, Sep. 2020, pp. 127–141, doi: 10.1007/978-3-030-59638-5_9.

[74] C. Klinkmüller, A. Ponomarev, A. B. Tran, I. Weber, and W. Aalst, ''Mining blockchain processes: Extracting process mining data from blockchain applications,'' in *Proc. Int. Conf. Bus. Process Manag.*, Cham, Switzerland: Springer Verlag, 2019, pp. 71–86, doi: 10.1007/978-3-030-30429-4_6.

[75] R. K. L. Ko, S. S. G. Lee, and E. Wah Lee, ''Business process management (BPM) standards: A survey,'' *Bus. Process Manag. J.*, vol. 15, no. 5, pp. 744–791, Sep. 2009, doi: 10.1108/14637150910987937.

[76] T. Koens and E. Poll. (Sep. 2018). *What Blockchain Alternative Do You Need?*. [Online]. Available: https://doi.org/10.1007/978-3-030-00305-0_9

[77] J. Ladleif, M. Weske, and I. Weber, ''Modeling and enforcing blockchain-based choreographies,'' in *Proc. Int. Conf. Bus. Process Manag.* (Lecture Notes in Computer Science). Berlin, Germany: Springer Verlag, 2019, pp. 69–85, doi: 10.1007/978-3-030-26619-6_7.

[78] C. Lauster, P. Klinger, N. Schwab, and F. Bodendorf, ''Literature review linking blockchain and bussiness process management,'' in *Proc. 15th Int. Conf. Wirtschaftsinformatik, Potsdam*, 2020, pp. 1802–1817, doi: 10.30844/wi_2020_r10-klinger.

[79] A. F. Leonardo Leite, G. A. Oliva, M. G. Nogueira, M. A. Gerosa, F. Kon, and S. D. Milojicic, ''A systematic literature review of service choreography adaptation,'' *Serv. Oriented Comput. Appl.*, vol. 7, no. 3 pp. 199–216, Sep. 2013, doi: 10.1007/s11761-012-0125-z.

[80] A. Lohachab, S. Garg, B. Kang, M. B. Amin, J. Lee, S. Chen, and X. Xu, ''Towards interconnected blockchains,'' *ACM Comput. Surv.*, vol. 54, no. 7, pp. 1–39, Sep. 2022, doi: 10.1145/3460287.

[81] O. Lopez-Pintado, M. Dumas, L. Garccia-Banuelos, and I. Weber, "Dynamic role binding in blockchain-based collaborative bussiness processes," in *Proc. Int. Conf. Adv. Inf. Syst. Eng.* (Lecture Notes in Computer Science). Cham, Switzerland: Springer Verlag, 2019, pp. 399–414, doi: 10.1007/978-3-030-21290-2_25.

[82] O. Lopez-Pintado, M. Dumas, L. Garcia-Banuelos, and I. Weber, "Interpreted execution of business process models on blockchain," in *Proc. IEEE 23rd Int. Enterprise Distrib. Object Comput. Conf. (EDOC)*, Oct. 2019, pp. 206–215, doi: 10.1109/EDOC.2019.00033.

[83] O. López-Pintado, M. Dumas, L. García-Bañuelos, and I. Weber, "Controlled flexibility in blockchain-based collaborative business processes," *Inf. Syst.*, vol. 104, Feb. 2022, Art. no. 101622, doi: 10.1016/j.is.2020.101622.

[84] O. López-Pintado, M. Dumas, L. García-Bañuelos, and I. Weber. (2017). *Caterpillar: A Blockchain-Based Business Process Management System*. [Online]. Available: https://www.researchgate.net/publication/329587092_Caterpillar_A_Blockchain-Based_Business_Process_Management_System

[85] O. López-Pintado, L. García-Bañuelos, M. Dumas, I. Weber, and A. Ponomarev, "Caterpillar: A business process execution engine on the Ethereum blockchain," *Softw. - Pract. Exp.*, vol. 49, no. 7 pp. 1162–1193, Jul. 2019, doi: 10.1002/spe.2702.

[86] R. Lu and S. Sadiq, "A survey of comparative business process modeling approaches," in *Proc. Int. Conf. Bus. Inf. Syst.* (Lecture Notes in Computer Science). Cham, Switzerland: Springer Verlag, 2007, pp. 82–94, doi: 10.1007/978-3-540-72035-5_7.

[87] M. Macdonald, L. Liu-Thorrold, and R. Julien. (2017). *The Blockchain: A Comparison of Platforms and their Uses Beyond Bitcoin*. [Online]. Available:https://www.ncbi.nlm.nih.gov/pmc/articles/PMC7787359/

[88] M. F. Madsen, "Collaboration among adversaries: Distributed workflow execution on a blockchain," in *Proc. Symp. Found. Appl. Blockchain*, Los Angeles, CA, USA, 018, pp. 8–15.

[89] P. D. Manuel, S. T. Selvi, and M. I. A.-E. Barr, "Trust management system for grid and cloud resources," in *Proc. 1st Int. Conf. Adv. Comput.*, Dec. 2009, pp. 176–181, doi: 10.1109/ICADVC.2009.5378187.

[90] J. Mattila. (2016). *ETLA Working Papers The Disruptive Potential of Distributed Consensus Architectures*. Accessed: Jun. 15, 2021. [Online]. Available: http://pub.etla.fi/ETLA-Working-Papers-38.pdf

[91] A. Mavridou and A. Laszka, "Designing secure ethereum smart contracts: A finite state machine based approach," in *Proc. Int. Conf. Financial Cryptogr. Data Secur.*, vol. 10957, Feb. 2018, pp. 523–540, doi: 10.1007/978-3-662-58387-6_28.

[92] A. Meidan, J. A. García-García, M. J. Escalona, and I. Ramos, "A survey on business processes management suites," *Comput. Standards Interfaces*, vol. 51, pp. 71–86, Mar. 2017, doi: 10.1016/j.csi.2016.06.003.

[93] A. Meironke, T. Seyffarth, and J. Damarowsky, "Business process compliance and blockchain: how does the ethereum blockchain address challenges of bus. process compliance?" In *Proc. 14th Int. Conf. Wirtschaftsinformatik (WI)*, Siegen, Germany, 2019, pp. 45–85.

[94] Jan Mendling, *Towards Blockchain Support for Business Processes*. New York, NY, USA: Springer Verlag, 2015, pp. 243–248, doi: 10.1007/978-3-319-94214-8_15.

[95] J. Mendling, B. Baesens, A. Bernstein, and M. Fellmann. (2017). *Challenges of Smart Business Process Management: An Introduction to the Special Issue. Decision Support Systems*. [Online]. Available: https://doi.org/10.1016/j.dss.2017.06.009

[96] J. Mendling, J. Brocke, M. Rosemann, M. Singh, T. Slaats, M. Staples, M. Weske, X. Xu, and L. Zhu, "Blockchains for business process management-challenges and opportunities," *ACM Trans. Manag. Inf. Syst.*, vol. 9, no. 1, p. 567, Feb. 2018, doi: 10.1145/3183367.

[97] J. Mendling, G. Decker, R. Hull, H. A. Reijers, and I. Weber, "How do machine learning, robotic process automation, and blockchains affect the human factor in business process management?" *Commun. Assoc. Inf. Syst.*, vol. 2, pp. 297–320, May 2018, doi: 10.17705/1CAIS.04319.

[98] L. Mercenne, K.-L. Brousmiche, and E. B. Hamida, "Blockchain studio: A role-based business workflows management system," in *Proc. IEEE 9th Annu. Inf. Technol., Electron. Mobile Commun. Conf. (IEMCON)*, Nov. 2018, pp. 1215–1220, doi: 10.1109/IEMCON.2018.8614879.

[99] G. Meroni and P. Plebani, "Combining artifact-driven monitoring with blockchain: Analysis and solutions," in *Proc. Int. Conf. Adv. Inf. Syst. Eng.* Cham, Switzerland: Springer Verlag, 2018, pp. 103–114, doi: 10.1007/978-3-319-92898-2_8.

[100] G. Meroni, P. Plebani, and F. Vona, "Trusted artifact-driven process monitoring of multi-party bussiness processes with blockchain," in *Proc. Int. Conf. Bus. Process Manag.* London, U.K.: Springer Verlag, 2019, pp. 55–70, doi: 10.1007/978-3-030-30429-4_5.

[101] F. Milani, L. Garcia-Banuelos, H. A. Reijers, and L. Stepanyan, "Business process redesign heuristics for blockchain solutions," in *Proc. IEEE 24th Int. Enterprise Distrib. Object Comput. Conf. (EDOC)*, Oct. 2020, pp. 209–216, doi: 10.1109/EDOC49727.2020.00033.

[102] V. J. Morkunas, J. Paschen, and E. Boon, "How blockchain technologies impact your business model," *Bus. Horizons*, vol. 62, no. 3, pp. 295–306, May 2019, doi: 10.1016/J.BUSHOR.2019.01.009.

[103] W. Mougayar and V. Buterin, *The Business Blockchain: Promise, Practice, and Application of the Next Internet Technology*. Hoboken, NJ, USA: Wiley, May 2021.

[104] R. Mühlberger, S. Bachhofner, C. Ciccio, L. García-Bañuelos, and O. Löpez-Pintado, "Extracting event logs for process mining from data stored on the blockchain," in *Proc. Int. Conf. Bus. Process Manag.*, 2019, pp. 690–703, doi: 10.1007/978-3-030-37453-2_55.

[105] M. Müller, N. Ostern, and M. Rosemann, "Silver bullet for all trust issues? Blockchain-based trust patterns for collaborative business processes," in *Proc. Int. Conf. Bus. Process Manag.*, vol. 393, Sep. 2020, pp. 3–18, doi: 10.1007/978-3-030-58779-6_1.

[106] H. Nakamura, K. Miyamoto, and M. Kudo, "Inter-organizational Bus. Processes Managed by Blockchain," in *Proc. Int. Conf.* Cham, Switzerland: Springer Verlag, 2018, pp. 3–17, doi: 10.1007/978-3-030-02922-7_1.

[107] A. Norta, L. Ma, Y. Duan, A. Rull, M. Kalvart, and K. Taveter, "EContractual choreography-language properties towards cross-organizational business collaboration," *J. Internet Services Appl.*, vol. 6, no. 1, pp. 1–23, Dec. 2015, doi: 10.1186/s13174-015-0023-7.

[108] F. Panduwinata and P. Yugopuspito, "BPMN approach in blockchain with hyperledger composer and smart contract: Reservation-based parking system," in *Proc. 5th Int. Conf. New Media Stud. (CONMEDIA)*, Oct. 2019, pp. 89–93, doi: 10.1109/CONMEDIA46929.2019.8981845.

[109] Christoph Prybila, Stefan Schulte, Christoph Hochreiner, and Ingo Weber, "Runtime verification for business processes utilizing the Bitcoin blockchain," *Futur. Gener. Comput. Syst.*, vol. 107, pp. 816–831, Jun. 2020, doi: 10.1016/j.future.2017.08.024.

[110] Z. M. Qiu and Y. S. Wong, "Dynamic workflow change in PDM systems," *Comput. Ind.*, vol. 58, no. 5, pp. 453–463, Jun. 2007, doi: 10.1016/j.compind.2006.09.014.

[111] M. M. Queiroz, R. Telles, and S. H. Bonilla, "Blockchain and supply chain management integration: A systematic review of the literature," *Supply Chain Manag., Int. J.*, vol. 25, no. 2, pp. 241–254, Aug. 2019, doi: 10.1108/SCM-03-2018-0143.

[112] S. D. Ramchurn, D. Huynh, and N. R. Jennings, "Trust in multi-agent systems," *Knowl. Eng. Rev.*, vol. 19, no. 1, pp. 1–25, 2004, doi: 10.1017/S0269888904000116.

[113] M. Reichert and B. Weber, *Enabling Flexibility in Process-Aware Information Systems: Challenges, Methods, Technologies*. Cham, Switzerland: Springer, 2012.

[114] P. Rimba, A. B. Tran, I. Weber, M. Staples, A. Ponomarev, and X. Xu, "Comparing blockchain and cloud services for business process execution," in *Proc. IEEE Int. Conf. Softw. Archit. (ICSA)*, Apr. 2017, pp. 257–260, doi: 10.1109/ICSA.2017.44.

[115] P. Rimba, A. B. Tran, I. Weber, M. Staples, A. Ponomarev, and X. Xu, "Quantifying the cost of distrust: Comparing blockchain and cloud services for business process execution," *Inf. Syst. Frontiers*, vol. 22, no. 2, pp. 489–507, Apr. 2020, doi: 10.1007/s10796-018-9876-1.

[116] M. La Rosa, H. A. Reijers, W. M. P. van der Aalst, R. M. Dijkman, J. Mendling, M. Dumas, and L. García-Bañuelos, "APROMORE: An advanced process model repository," *Expert Syst. Appl.*, vol. 38, no. 6, pp. 7029–7040, Jun. 2011, doi: 10.1016/j.eswa.2010.12.012.

[117] R. Rosa and C. E. Rothenberg, "Blockchain-based decentralized applications for multiple administrative domain networking," *IEEE Commun. Standards Mag.*, vol. 2, no. 3, pp. 29–37, Oct. 2018, doi: 10.1109/MCOMSTD.2018.1800015.

[118] P. Runeson and M. Höst, "Guidelines for conducting and reporting case study research in software engineering," *Empirical Softw. Eng.*, vol. 14, no. 2, pp. 131–164, Apr. 2009, doi: 10.1007/s10664-008-9102-8.

[119] N. Russell, W. M. P. Van Der Aalst, A. H. M. T. Hofstede, and D. Edmond, "Workflow resource patterns: Identification, representation and tool support," in *Proc. Int. Conf. Adv. Inf. Syst. Eng.*, Cham, Switzerland: Springer Verlag, 2005, pp. 216–232, doi: 10.1007/11431855_16.

[120] M. Schinle, C. Erler, P. N. Andris, and W. Stork, "Integration, execution and monitoring of business processes with chaincode," in *Proc. 2nd Conf. Blockchain Res. Appl. Innov. Netw. Services (BRAINS)*, Sep. 2020, pp. 63–70, doi: 10.1109/BRAINS49436.2020.9223283.

[121] S. Schmidt, M. Jung, T. Schmidt, I. Sterzinger, G. Schmidt, M. Gomm, T. Tschirschke, T. Reisinger, F. Schlarb, D. Benkenstein, and B. Emig, "Unibright-the unified framework for blockchain based business integration," in *Proc. Unibright Project White Paper*, 2018, pp. 189–199.

[122] S. Schulte, C. Janiesch, S. Venugopal, I. Weber, and P. Hoenisch, "Elastic business process management: State of the art and open challenges for BPM in the cloud," *Future Gener. Comput. Syst.*, vol. 46, pp. 36–50, May 2015, doi: 10.1016/j.future.2014.09.005.

[123] B. A. Scriber, "A framework for determining blockchain applicability," *IEEE Softw.*, vol. 35, no. 4, pp. 70–77, Jul./Aug. 2018, doi: 10.1109/MS.2018.2801552.

[124] D. Silva, S. Guerreiro, and P. Sousa, "Decentralized enforcement of business process control using blockchain," vol. 12, pp. 69–87, May 2018, doi: 10.1007/978-3-030-06097-8_5.

[125] N. Six, "Decision process for blockchain architectures based on requirements," 2021, *arXiv:2101.08998*.

[126] Y. Sompolinsky and A. Zohar, "Secure high-rate transaction processing in bitcoin," in *Proc. Int. Conf. Financial Cryptogr. Data Secur.*, Cham, Switzerland: Springer Verlag, 2015,pp. 507–527, doi: 10.1007/978-3-662-47854-7_32.

[127] C. Sturm, J. Scalanczi, S. Schönig, and S. Jablonski, "A blockchain-based and resource-aware process execution engine," *Future Gener. Comput. Syst.*, vol. 100, pp. 19–34, Nov. 2019, doi: 10.1016/j.future.2019.05.006.

[128] C. Sturm, J. Szalanczi, S. Stefan, and S. Jablonski, "A lean architecture for blockchain based decentralized process execution," in *Proc. Int. Conf. Bus. Process Manag.*, New York, NY, USA: Springer Verlag, 2019, pp. 361–373, doi: 10.1007/978-3-030-11641-5_29.

[129] H. Tang, Y. Shi, and P. Dong, "Public blockchain evaluation using entropy and TOPSIS," *Expert Syst. Appl.*, vol. 117, pp. 204–210, Mar. 2019. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0957417418306250

[130] A. B. Tran, Q. Lu, and I. Weber, "Lorikeet: A model-driven engineering tool for blockchain-based business process execution and asset management," in *Proc. Int. Conf. Bus. Process Manag.*, Sydney, NSW, Australia, 2018, pp. 56–60.

[131] Ching Hong Tsai, Kuo Chan Huang, Feng Jian Wang, and Chun Hao Chen. 2010, "A distributed server architecture supporting dynamic resource provisioning for BPM-oriented workflow management systems," *J. Syst. Softw.*, vol. 83, no. 8, pp. 1538–1552, Aug. 2010, doi: 10.1016/j.jss.2010.04.001.

[132] V. A. de Sousa and B. Corentin, "Towards an integrated methodology for the development of blockchain-based solutions supporting cross-organizational processes," in *Proc. 13th Int. Conf. Res. Challenges Inf. Sci. (RCIS)*, May 2019, doi: 10.1109/RCIS.2019.8877045.

[133] W. Viriyasitavat, "Multi-criteria selection for services selection in service workflow," *J. Ind. Inf. Integr.*, vol. 1, pp. 20–25, Mar. 2016, doi: 10.1016/j.jii.2016.03.003.

[134] W. Viriyasitavat, T. Anuphaptrirong, and D. Hoonsopon, "When blockchain meets Internet of Things: Characteristics, challenges, and business opportunities," *J. Ind. Inf. Integr.*, vol. 15, pp. 21–28, Sep. 2019, doi: 10.1016/J.JII.2019.05.002.

[135] W. Viriyasitavat and Z. Bi, "Service selection and workflow composition in modern business processes," *J. Ind. Inf. Integr.*, vol. 17, Mar. 2020, Art. no. 100126, doi: 10.1016/j.jii.2020.100126.

[136] W. Viriyasitavat and D. Hoonsopon, "Blockchain characteristics and consensus in modern business processes," *J. Ind. Inf. Integr.*, vol. 13, pp. 32–39, Mar. 2019, doi: 10.1016/J.JII.2018.07.004.

[137] W. Viriyasitavat and A. Martin, "Formal trust specification in service workflows," in *Proc. IEEE/IFIP Int. Conf. Embedded Ubiquitous Comput.*, Dec. 2010, pp. 703–710, doi: 10.1109/EUC.2010.111.

[138] W. Viriyasitavat and A. Martin, "In the relation of workflow and trust characteristics, and requirements in service workflows," in *Communications in Computer and Information Science*. Berlin, Germany: Springer, 2011, pp. 492–506, doi: 10.1007/978-3-642-25327-0_42.

[139] W. Viriyasitavat and A. Martin, "A survey of trust in workflows and relevant contexts," *IEEE Commun. Surveys Tuts.*, vol. 14, no. 3, pp. 911–940, 3rd Quart., 2012, doi: 10.1109/SURV.2011.072811.00081.

[140] W. Viriyasitavat, L. D. Xu, and Z. Bi, "RmSWSpec: Real-time monitoring of service workflow specification language for specification patterns," *IEEE Trans. Ind. Informat.*, vol. 15, no. 7, pp. 4021–4032, Jul. 2019, doi: 10.1109/TII.2018.2889749.

[141] W. Viriyasitavat, L. D. Xu, and Z. Bi, "Specification patterns of service-based applications using blockchain technology," *IEEE Trans. Computat. Soc. Syst.*, vol. 7, no. 4, pp. 886–889, Aug. 2020, doi: 10.1109/TCSS.2020.2999574.

[142] W. Viriyasitavat, L. D. Xu, Z. Bi, and D. Hoonsopon, "Blockchain technology for applications in Internet of Things—Mapping from system design perspective," *IEEE Internet Things J.*, vol. 6, no. 5, pp. 8155–8168, Oct. 2019, doi: 10.1109/JIOT.2019.2925825.

[143] W. Viriyasitavat, L. D. Xu, Z. Bi, D. Hoonsopon, and N. Charoenruk, "Managing QoS of Internet-of-Things services using blockchain," *IEEE Trans. Computat. Social Syst.*, vol. 6, no. 6, pp. 1357–1368, Dec. 2019, doi: 10.1109/TCSS.2019.2919667.

[144] W. Viriyasitavat, L. D. Xu, Z. Bi, and V. Pungpapong, "Blockchain and Internet of Things for modern business process in digital economy—The state of the art," *IEEE Trans. Computat. Social Syst.*, vol. 6, no. 6, pp. 1420–1432, Dec. 2019, doi: 10.1109/TCSS.2019.2919325.

[145] W. Viriyasitavat, L. Xu, Z. Bi, and A. Sapsomboon, "Blockchain-based business process management (BPM) framework for service composition in industry 4.0," *J. Intell. Manuf.*, vol. 26, pp. 1–12, May 2018, doi: 10.1007/s10845-018-1422-y.

[146] W. Viriyasitayat, L. Da Xu, Z. Bi, and A. Sapsomboon, "Extension of specification language for soundness and completeness of service workflow," *Enterprise Inf. Syst.*, vol. 12, no. 5, pp. 638–657, Jan. 2018, doi: 10.1080/17517575.2018.1432769.

[147] W. Viriyasitavat, L. Da Xu, Z. Bi, and A. Sapsomboon, "New blockchain-based architecture for service interoperations in Internet of Things," *IEEE Trans. Computat. Social Syst.*, vol. 6, no. 4, pp. 739–748, Aug. 2019, doi: 10.1109/TCSS.2019.2924442.

[148] W. Viriyasitavat, L. D. Xu, and A. Martin, "SWSpec: The requirements specification language in service workflow environments," *IEEE Trans. Ind. Informat.*, vol. 8, no. 3, pp. 631–638, Aug. 2012, doi: 10.1109/TII.2011.2182519.

[149] W. Viriyasitavat, L. D. Xu, and W. Viriyasitavat, "Compliance checking for requirement-oriented service workflow interoperations," *IEEE Trans. Ind. Informat.*, vol. 10, no. 2, pp. 1469–1477, May 2014, doi: 10.1109/TII.2014.2301132.

[150] Y. Wang, J. H. Han, and P. Beynon-Davies, "Understanding blockchain technology for future supply chains: A systematic literature review and research agenda," *Supply Chain Manag.*, vol. 24, no. 1, pp. 62–84, Jan. 2019, doi: 10.1108/SCM-03-2018-0148.

[151] I. Weber, X. Xu, R. Riveret, G. Governatori, A. Ponomarev, and J. Mendling, "Using blockchain to enable untrusted business process monitoring and execution," Tech. Rep. UNSW-CSE-TR-09, 2016.

[152] I. Weber, X. Xu, and R. Riveret, "Untrusted business process monitoring and execution using blockchain," in *Proc. Int. Conf. Bus. Process Manag.* Cham, Switzerland: Springer Verlag, 2016, pp. 329–347, doi: 10.1007/978-3-319-45348-4_19.

[153] B. Wetzstein, D. Karastoyanova, O. Kopp, F. Leymann, and D. Zwink, "Cross-organizational process monitoring based on service choreographies," in *Proc. ACM Symp. Appl. Comput. (SAC)*, 2010, pp. 2485–2490, doi: 10.1145/1774088.1774601.

[154] K. Wust and A. Gervais, "Do you need a blockchain?" in *Proc. Crypto Valley Conf. Blockchain Technol. (CVCBT)*, Jun. 2018, pp. 45–54, doi: 10.1109/CVCBT.2018.00011.

[155] L. Xu and W. Viriyasitavat, "Application of blockchain in collaborative Internet-of-Things services," *IEEE Trans. Computat. Soc. Syst.*, vol. 6, no. 6, pp. 1295–1303, May 2019.

[156] B. Yu, X. Lu, H. Chen, M. Lei, and X. Wang, *Runtime Verification of Ethereum Smart Contracts Based on MSVL*. Cham, Switzerland: Springer, 2021, pp. 142–153, doi: 10.1007/978-3-030-77474-5_10.

[157] P. Zhang and M. Zhou, "Security and trust in blockchains: Architecture, key technologies, and open issues," *IEEE Trans. Computat. Social Syst.*, vol. 7, no. 3, pp. 790–801, Jun. 2020, doi: 10.1109/TCSS.2020.2990103.

[158] Z. Bi, L. D. Xu, and C. Wang, "Internet of Things for enterprise systems of modern manufacturing," *IEEE Trans. Ind. Informat.*, vol. 10, no. 2, pp. 1537–1546, May 2014, doi: 10.1109/TII.2014.2300338.

[159] (Apr. 19, 2021). *Standard for the Framework of Blockchain Use in Internet of Things (IoT)—IEEE P2418.1 Working Group*. [Online]. Available: https://sagroups.ieee.org/2418-1/

[160] (Apr. 19, 2021). *ISO—ISO/TC 307—Blockchain and Distributed Ledger Technologies*. [Online]. Available: https://www.iso.org/committee/6266604.html

[161] (May 14, 2021). *DIF–Decentralized Identity Foundation*. [Online]. Available: https://identity.foundation/

[162] (May 14, 2021). *ID2020: Digital Identity With Blockchain | Accenture*. [Online]. Available: https://www.accenture.com/us-en/insight-blockchain-id2020

[163] (May 13, 2021). *A Comprehensive List of Cryptocurrency Exchange Hacks—SelfKey*. [Online]. Available: https://selfkey.org/list-of-cryptocurrency-exchange-hacks/

[164] (May 13, 2021). *What's Up with Blockchain Interoperability? | Coinzilla Academy*. [Online]. Available: https://academy.coinzilla.com/blockchain-interoperability/

[165] (May 13, 2021). *Blockchain Industrial Alliance*. [Online]. Available: https://bialliance.io/

[166] Z. Bi, Y. Jin, P. Maropoulos, W.-J. Zhang, and L. Wang, "Internet of Things (IoT) and big data analytics (BDA) for digital manufacturing (DM)," *Int. J. Prod. Res.*, pp. 1–18, Jul. 2021, doi: 10.1080/00207543.2021.1953181.

[167] Z. Bi, C. W. J. Zhang, C. Wu, and L. Li, "New digital triad (DT-II) concept for lifecycle information integration of sustainable manufacturing systems," *J. Ind. Inf. Integr.*, vol. 26, Mar. 2022, Art. no. 100316.

[168] Z. M. Bi, L. D. Xu, and P. R. Ouyang, *Smart Manufacturing—Theories, Methods, and Application. Machines*, vol. 10. Cham, Switzerland: Springer, 2022, p. 742.

[169] T. R. Gadekallu, Q.-V. Pham, D. C. Nguyen, P. K. R. Maddikunta, N. Deepa, B. Prabadevi, P. N. Pathirana, J. Zhao, and W.-J. Hwang, "Blockchain for edge of things: Applications, opporuntities, and challenges," *IEEE Internet Things J.*, vol. 9, no. 2, pp. 964–988, 2022.

[170] Y. Zhao, X. Yang, Y. Yu, B. Qin, X. Du, and M. Guizani, "Blockchain-based auditable privacy-preserving data classification for Internet of Things," *IEEE Internet Things J.*, vol. 9, no. 4, pp. 2468–2484, Feb. 2022.

[171] H. Cheng, Q. Hu, X. Zhang, Z. Yu, Y. Yang, and N. Xiong, "Trusted resource allocation based on smart contracts for blockchain-enabled Internet of Things," *IEEE Internet Things J.*, vol. 9, no. 11, pp. 7904–7915, Jun. 2022.

[172] S. Ahmed, M. A. Shah, and K. Wakil, "Blockchain as a trust builder in the smart city domain: A systematic literature review," *IEEE Access*, vol. 8, pp. 92977–92985, 2020.

[173] S. Siddiqui, S. Hameed, S. A. Shah, I. Ahmad, A. Aneiba, D. Draheim, and S. Dustdar, "Toward software-defined networking-based IoT frameworks: A systematic literature review, taxonomy, open challenges and prospects," *IEEE Access*, vol. 10, pp. 70850–70901, 2022.

[174] A. Belapurkar, *Distributed Systems Security: Issues, Processes and Solutions*. Hoboken, NJ, USA: Wiley, 2009.

[175] J. S. Kowalik, J. Gorski, and A. Sachenko, "Cyberspace security and defense: research issues," in *Proc. NATO Adv. Res. Workshop Cyberspace Sequrity Defense, Reserarch*, Gdansk, Poland, Sep. 2004, pp. 78–89.

[176] W. Viriyasitavat, L. D. Xu, G. Dhiman, A. Sapsomboon, V. Pungpapong, and Z. Bi, "Service workflow: State-of-the-art and future trends," *IEEE Trans. Services Comput.*, early access, May 12, 2021, doi: 10.1109/TSC.2021.3121394.

WATTANA VIRIYASITAVAT (Senior Member, IEEE) received the M.S. degree in computer engineering from Carnegie Mellon University, Pittsburgh, PA, USA, in 2006, and the D.Phil. (Ph.D.) degree in computer science from the University of Oxford, in 2013. He is currently an Assistant Professor with the Business Information Technology Division, Department of Statistics, Faculty of Commerce and Accountancy, Chulalongkorn University, Thailand. His research interests include trust, security in service workflows, business process, specification language, blockchain, the Internet of Things, and compliance checking.

LI DA XU (Fellow, IEEE) received the B.S. and M.S. degrees in information science from the University of Science and Technology of China, in 1978 and 1981, respectively, and the Ph.D. degree in systems science and engineering from Portland State University, Portland, OR, USA, in 1986. He is an Academician of the Russian Academy of Sciences (formerly the Academy of Sciences of USSR), the Russian Academy of Engineering (formerly the Academy of Engineering of USSR), and the European Academy of Sciences (Division of Engineering). From 2016 to 2021, he was a Highly Cited Researcher in the field of engineering named by Clarivate Analytics (formerly Thomson Reuters Intellectual Property and Science).

DUSIT NIYATO (Fellow, IEEE) received the B.Eng. degree from the King Mongkut's Institute of Technology Ladkrabang, in 1999, and the Ph.D. degree in electrical and computer engineering from the University of Manitoba, Winnipeg, MB, Canada, in 2008. He is currently a Full Professor with the School of Computer Engineering, Nanyang Technological University, Singapore. His research interests include green communication, the Internet of Things, and sensor networks.

ZHUMING BI (Senior Member, IEEE) received the dual Ph.D. degree from the Harbin Institute of Technology, Harbin, China, and the University of Saskatchewan, Saskatoon, SK, Canada, in 1994 and 2002, respectively. He is currently the Harris Chair in wireless communications and applied research and a Professor with the Department of Civil and Mechanical Engineering, Purdue University Fort Wayne (PFW). He has authored or coauthored four books, including three of them on *Finite Element Analysis Applications*, *Computer Aided Design and Manufacturing*, and *Digital Manufacturing*. In addition, he has published ten book chapters, 155 journal articles, and over 60 articles in conference proceedings. His research interests include the Internet of Things, enterprise systems, digital manufacturing, finite element analysis, machine designs, and robotics and automation.

DANUPOL HOONSOPON received the D.B.A. degree in marketing from Chulalongkorn University, Bangkok, Thailand, in 2010. He is currently an Assistant Professor with the Chulalongkorn Business School, Chulalongkorn University. He has authored or coauthored in several journals, such as *Journal of Industrial Information Integration*, *Journal of Managerial Issues*, *Australasian Marketing Journal*, and *Journal of Asia-Pacific Business*. His current research interests include teaching new product development, innovation management, and marketing research.