Ngày gửi bài: 30/03/2022

Họ và tên tác giả: Hà Ngọc Long[1], Đỗ Sông Hương[2]

- **Tác giả chính, liên hệ**: Hà Ngọc Long

Học hàm và học vị: Tiến sĩ

Tổ chức công tác:

- [1]Khoa Hệ thống Thông tin Kinh tế, trường Đại học Kinh tế, Đại học Huế
- [2]Khoa Kế toán Tài chính, trường Đại học Kinh tế, Đại học Huế

Thông tin liên lạc của tác giả liên hệ:

- Địa chỉ email: hnlong@hueuni.edu.vn
- Điện thoại: 0935.626.575

Lĩnh vực nghiên cứu sâu của tác giả chính: Quản trị quy trình kinh doanh, tuân thủ nguyên tắc và pháp lý trong tài chính (BPM, Compliance & RegTech); Chuyển đổi số; Tiền điện tử và hợp đồng thông minh trong công nghệ tài chính; Khai phá quy trình kinh doanh; Microservices.

Tên bài viết: Đề Xuất Mô Hình Quản Trị Tuân Thủ Quy Trình Dựa Trên Nền Tảng Điện Toán Đám Mây

Chủ đề: Chuyển đổi số trong các doanh nghiệp trong bối cảnh đại dịch Covid-19

Lĩnh vực nghiên cứu của bài viết: Hệ thống Thông tin, Quản trị Quy trình Nghiệp vụ.

Số từ của bài viết: 7525 từ

Một số tạp chí tiêu biểu mà tác giả chính đã từng đăng tải công trình nghiên cứu:

- Business Process Management Journal (SSCI/Scopus - Q1)
- IEEE Access (SCIE - Q1)

Lời cam kết:

Nhóm tác giả cam kết rằng: bài viết thuộc bản quyền hợp pháp của nhóm tác giả và chưa từng được công bố trên bất kỳ tạp chí nào trước đó. Bài báo này chỉ được gửi cho Hội thảo quốc tế "*Phục hồi và phát triển kinh tế Việt Nam trong bối cảnh đại dịch Covid-19*" bình duyệt và không được đồng thời gửi cho tạp chí nào khác.

Tên bài viết: ĐỀ XUẤT MÔ HÌNH QUẢN TRỊ TUÂN THỦ QUY TRÌNH DỰA TRÊN NỀN TẢNG ĐIỆN TOÁN ĐÁM MÂY

Tác giả:

Hà Ngọc Long – Khoa Hệ Thống Thông Tin Kinh Tế, Đại học Kinh Tế, Đại học Huế

Đỗ Sông Hương – Khoa Kế Toán – Tài Chính, Đại học Kinh Tế, Đại học Huế

Tóm tắt

Nhiều tổ chức và tập đoàn phải đối mặt với áp lực ngày càng tăng trong việc thích ứng nhanh chóng và linh hoạt với môi trường kinh doanh. Những giải pháp phần mềm quản lý quy trình nghiệp vụ đã được sử dụng để quản lý các quy trình nghiệp vụ của doanh nghiệp. Do đó, nhu cầu kiểm tra tuân thủ (BPC) trong các quy trình kinh doanh ngày càng trở nên cấp thiết. Microservices (MSA) nổi lên như là một kiến trúc tiềm năng cho điện toán đám mây. Kiến trúc này được cho là sẽ làm thay đổi cách các ứng dụng đám mây được xây dựng. Bài báo này trước tiên sẽ giới thiệu ngắn gọn các nghiên cứu liên quan nhằm giải quyết các vấn đề tuân thủ đối với các quy trình kinh doanh. Dựa trên những lỗ hổng nghiên cứu được chỉ ra bởi các nghiên cứu liên quan, bài báo này đề xuất một mô hình tuân thủ quy định được thiết kế dành riêng cho microservices, sử dụng các công nghệ tiên tiến trong nghiên cứu tuân thủ quy trình kinh doanh. Mô hình này được đặt tên là BPC4MSA. Bằng cách trình bày BPC4MSA, bài báo thảo luận về các cơ hội và thách thức của việc áp dụng kiến trúc microservices, mà có thể hữu ích cho các nhà nghiên cứu và thực hành.

Từ khóa: Điện toán đám mây, Giao diện lập trình ứng dụng, Microservices, Quản lý quy trình nghiệp vụ, Tuân thủ quy trình nghiệp vụ.

Realization of Modern Cloud-Based Business Process Compliance Architecture

N. Long Ha[1] and S. Huong Do[2]

[1] Faculty of Economic Information Systems, University of Economics, Hue University
hnlong@hueuni.edu.vn
[2] Faculty of Accounting and Finance, University of Economics, Hue University
dshuong@hueuni.edu.vn

Abstract: Many organizations and corporations face growing pressure to rapidly and flexibly adapt to the agile business environment. As a result, they have employed business process management software to manage their business processes in a process-oriented environment. Thus, the demand for compliance in business processes is increasing. The rise of microservices as architectural patterns for cloud computing creates many exciting opportunities for software architecture. This paper will first briefly introduce the preliminaries and investigate the related studies that tackle the compliance issues for business processes. Based on the research gaps indicated by the structural literature review, this paper proposes a regulatory compliance framework designed explicitly for microservice, named BPC4MSA, that employs cutting-edge technologies in business process compliance research. By presenting BPC4MSA, the paper discusses the opportunities and challenges of adopting microservice architecture, which could be useful for prospective researchers and practitioners.

Keywords: Business process management, Business process compliance, Cloud computing, Microservices, Application programming interface.

1. Introduction

Vietnam recently approved the National Digital Transformation Programme by 2025, with a vision toward 2030. The initiative plans to accelerate the digital shift through changes in awareness, enterprise strategies, and incentives towards digitalizing businesses, administration, and production activities. Digital Transformation (DX) describes the practice of organizations using technology to automate and optimize their business processes and enhance customer experience. As a result, organizations stay competitive and relevant in the new customer-centric economy and help achieve their business goals (Binh and Phuong, 2020).

In a process-oriented business, organizations focus on creating, optimizing, and monitoring their business processes so that organizations can decrease cost, increase revenue, and boost operational efficiency. *Business Process Management* (BPM) discipline has emerged as a holistic technology to control and operate the entire business through rules that clearly define business processes so that these processes can be managed systematically. BPM helps create a competitive advantage by improving organizational agility in the long run. BPM is about continuous improvement and optimizing processes to ensure high performance and achieve agility and flexibility as a tool to gain competitive advantage (Møller *et al.*, 2007). A key benefit of BPM is the ability to adjust business processes accordingly to changing market requirements since business processes are dynamic, allowing businesses to respond more quickly and cost-effectively to changing market conditions (Hill *et al.*, 2006). Many organizations have already integrated BPM software solutions into their information technology (IT) infrastructure. The popular BPM solutions (BPMS) are often hosted in the cloud, offering more flexibility and cost reduction. In addition, business processes are subjected to be regulated. Thus, ensuring organizations' business processes comply with regulations or laws is essential for any BPMS (Kharbili *et al.*, 2008). The term *Business Process Compliance* (BPC) was coined to describe the alignment of business processes with relevant laws, regulations, and guidelines (Governatori and Sadiq, 2008). Within the scope of this work, the paper aims to focus on design-time business process compliance.

One of the key pillars of digital transformation must include cloud computing technology. DX by cloud computing platform helps build an ecosystem so that Vietnamese cloud computing businesses can master technology, provide standard cloud computing infrastructure and services to support and promote Vietnamese companies to DX and recover better in the post-COVID-19 era (Binh and Phuong, 2020). The next wave of digital technologies – cloud computing together with AI, blockchain, Internet of Things, and cloud-based services – has the potential to transform Vietnam into Asia's next high-performing economy and to bring up the living standards of all of Vietnam's citizens over the coming decades (Cameron *et al.*, 2019). Cloud computing can also increase energy efficiency, as data centers can better optimize performance with energy use and incorporate renewable energy and energy storage devices.

Recent studies reveal that the traditional business environment has been considered more than the cloud environment (Mustapha *et al.*, 2020). In addition, most cloud-based BPC approaches opt for service-oriented architecture (SOA), which is considered an *"outdated"* technology compared to the new type of design pattern, the microservice architecture (Richardson, 2018). Microservices are the next evolution of design architecture for cloud computing. The principle of microservices states that the application should be decomposed into small independent building blocks (the microservices), each focused on a single business capability that it is doing well. Each microservice communicates with the other using the application programming interfaces (APIs), and they can be updated independently. More research is needed on business process compliance centered on the cloud environment, especially the work based on microservices architecture. However, microservice is not a silver bullet to resolve all the problems. There are some cons and challenges when adopting microservice architecture (Messina *et al.*, 2016). One of the biggest challenges in microservices-based systems is partitioning into separated services. They should be simple enough to have a small set of responsibilities. Data management requires special attention because it can be one of the system's bottlenecks. So, conveniently, only one or few microservices access the data, but this can affect the responsiveness of the whole system. Given these settings, the main contributions of the paper are as follows:

1) It introduces a series of concepts related to the business process compliance topics, including BPM, BPC, microservices, and cloud technologies.
2) It carries out structured literature reviews of studies related to the business process compliance approaches across business environments. The review is not only aimed to provide a summary of existing evidence but to hint at the research gaps that need to be fulfilled.
3) It proposes the microservice architecture for business process compliance's use case, which facilitates the utilization of the BPC framework for very agile, real-world scenarios and a highly dynamic cloud-based environment.

The remainder of the paper is structured as follows. Section 2 introduces the basic concepts of BPM, BPC, microservices, and cloud computing. The literature view on the BPC frameworks and tools is provided in Section 3. Section 4 proposes a cloud-based microservice BPC architecture. Finally, Section 5 concludes our work.

2. Background

2.1. Business Process Management

In the process-oriented business environment, business processes are the backbone of an organization or business to ensure smooth and efficient operation. As business keeps growing, the need for information systems to manage their business processes becomes more urgent than ever. As a result, business process management (BPM) emerged to support operational efficiency and optimization in any organization (Dumas *et al.*, 2013).

BPM oversees how work is performed in an organization to assure consistent outcomes and take advantage of improvement opportunities (Dumas *et al.*, 2013). Thus, the critical idea of BPM is to focus on the identification, integration, optimization, implementation, monitoring, and management of processes. Fig. 1 shows the so-called BPM lifecycle, which describes a circular of designing to executing and monitoring phases to be repeated continuously. The lifecycle of BPM starts with the process identification phase. In this phase, a business problem is addressed and identified. Next, the business process is modeled in the process discovery phase. Then, the process analysis phase analyses the quality of the given business process. If the previous phase identifies an issue, the process redesign will address the problems known to introduce the to-be process model. Then, this model needs to be executed in the IT systems, which need to be developed to support this process, referred to as process automation in the implementation phase. Finally, when the process is executed, relevant data are collected and analyzed to evaluate the performance of the business process. When new issues arise, the cycle will need to be repeated continuously.

The success of any BPM project mainly comes from a good business process design. Without good design, any running system will eventually fail sooner or later. Thus, within the scope of this work, the paper will focus on the business process models in the designing phase as one of the main research subjects.

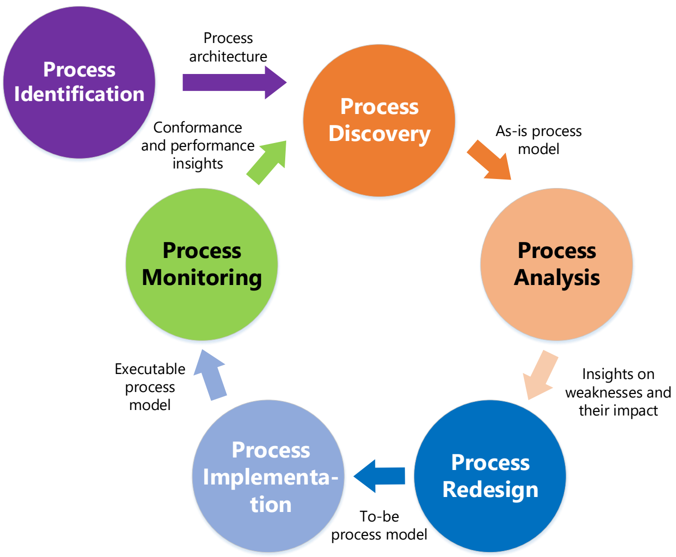

Fig. 1. The BPM lifecycle (Dumas *et al.*, 2013)

2.2. Business Process Compliance

Business processes are expected to comply with the increasing number of applicable laws and regulations imposed by the different origin and regulatory bodies. Ensuring business process compliance has been a central concern of any organization, especially after the 2008 financial crisis. However, the legal obligation of adhering to the regulation is costly in terms of time and money. As a result, many organizations turned to technology to digitalize the process of compliance. A new industry, a spin-off from fintech, is known as regulation technologies (RegTech). Regtech brings together different disciplines ranging from legal theory to computer science.

Compliance is the "*act/process to ensure that business operations, processes, and practices are in accordance with prescriptive (often legal) documents*" (Governatori, 2005). On the one hand, business processes are a primary criterion for accomplishing a business objective (López *et al.*, 2020). The business process compliance (BPC) topic also acquired increased attention from researchers and practitioners alongside the development of BPM and RegTech. Business process compliance by design in the course of ensuring the set of traces of a given process model that describes the process' execution behavior to be correctly executed as designed. Thus, BPC could be seen as a part of RegTech or the intersection between BPM and RegTech.

The business process compliance framework (Ha *et al.*, 2021) is described in Fig. 2. The framework shows the interactions between three spaces: (1) The regulation frameworks; (2) the business process compliance automation; and (3) the process-aware information systems (PAIS). The core of the BPC framework is the compliance checking engine which collects and computes the compliance based on the regulation specifications from the first space and process model from the third space. The distinct feature of this framework is that it employs a new method called *Partitioning Behavioral Retrieval* (PBR) to achieve computation efficiency (Ha and Prinz, 2021).

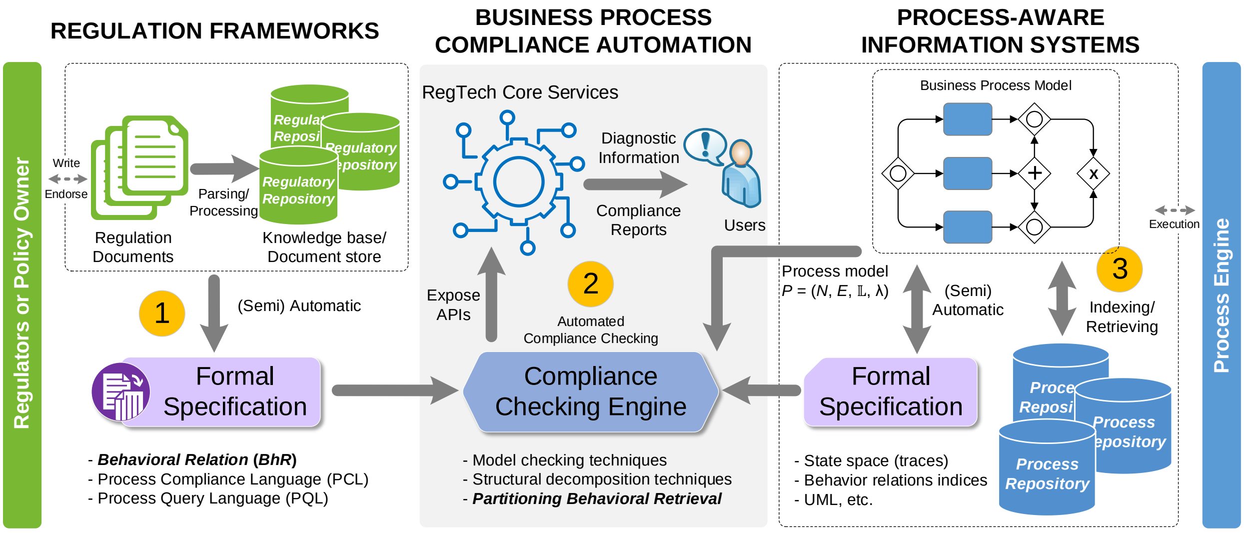

Fig. 2. Business process compliance framework (Ha *et al.*, 2021)

In today's ever-changing business environment, the BPC solutions employed in organizations need to quickly adapt to changes in the business environment, such as customer demands and dynamic changes in regulations (Hashmi *et al.*, 2018). Thus scalability and agility are the keys for a business to gain competitiveness. While scalability refers to increasing the capability to serve more customers, such as how easy it is to expand the IT infrastructure when needed, agility is a characteristic of how rapidly and easily a system can react to changes and be reorganized, reconfigured. The current BPC framework (Ha *et al.*, 2021) has failed in terms of scalability and agility since its primary concern is how to check business processes' compliance efficiently. The modern cloud-based BPC architecture could be a solution to resolve the drawbacks mentioned earlier. The paper will further discuss how modern cloud technologies can improve the current BPC framework in the following sections.

2.2. Microservices

Building *monolithic* applications is the beginning of software architecture. This kind of application includes all the application components, such as the user interface, business logic, and data access, into a single large codebase. Thus, the application is not decoupled by nature, which means each module or component does not perform its tasks independently. Monolithic is a traditional method of building applications, and in reality, many businesses still benefit from using it.

Nowadays, most IT applications are distributed due to highly available communication networks. Utilizing the monolithic application in the distributed environment becomes difficult, especially when the application grows more prominent. In the rapidly changing business environment, the monolithic architecture seems to fail to adapt to fast changes due to certain limitations: (1) limited scalability, (2) complex overtime, (3) difficulty to scale, (4) technology lock-in, and availability issue (Dragoni *et al.*, 2017).

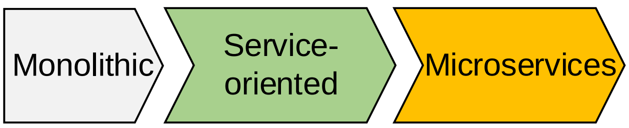

Fig. 3. Evolution of Software Architectures

To quickly respond to market conditions, businesses' applications require continuous updating and scaling up some small areas with zero downtime. The *microservices* architecture style has been proposed to cope with the problems mentioned above. Microservices represent the decomposition of monolithic systems into independently deployable services that do "*one thing well*" (Kratzke and Quint, 2017). In fact, microservices are the second iteration of the concept of *Service-Oriented Architecture* (SOA). The SOA, the first generation of service-oriented design patterns, typically uses heavyweight technologies such as SOAP (*Simple Object Access Protocol*) and other WS* (*Web Service**) standards (Richardson, 2018). They often use an *Enterprise Service Bus* (ESB), a middleware messaging component that contains business and message-processing logic to integrate the services, as the heart of the application. The rise of microservices is due to the increased complexity and introduced bottlenecks of SOA (Richardson, 2018). Microservices architecture emphasizes maximizing the decoupling of applications into smaller, independent components. As a result, it (1) enables highly customizable and easy to maintain; (2) ensures better uptime; (3) enables services to be independently scalable, the continuous delivery and deployment of large and complex applications; (4) enables developers' teams to be autonomous and technology stack independence; (5) allows to be easy experimenting and adoption of new technologies; and (6) has better fault detection, isolation, and recovery (Dragoni *et al.*, 2017; Richardson, 2018). However, there still exist several significant drawbacks and issues to this architecture, such as (1) challenges on how to systematically break an application into smaller pieces; (2) difficulties in developing, testing, and deploying distributed systems; (3) stringency in the coordination of deploying features that span multiple services (Richardson, 2018). Besides, organizations can embrace digital transformation initiatives more quickly by adopting microservices as part of their cloud migration with cloud service providers such as Microsoft Azure, Google Cloud Platform, Amazon Web Service, etc.

Unlike SOA, which often uses the ESB to support communication between services, the microservices use event bus for communication to create applications from services that are decoupled and as cohesive as possible (Alpers *et al.*, 2015). As a result, processing only happens inside each microservice, while information exchange between services relies on simple communication mechanisms (such as REST or messaging over a lightweight message bus) (Alpers *et al.*, 2015).

The increased size of organizations (in technology and number of employees) globally has led to applications with increased functionalities, which has escalated operational and server workloads. As a result, many organizations have already deployed their core system using the microservices architecture, such as Amazon, Best Buy, eBay, Netflix, Spotify, and Uber. Since organizations might already apply a microservices mindset in their whole IT systems, implementing a BPC application based on microservices architecture will simplify the integration of BPC into the existing organization system, thus saving time and reducing the cost of implementation. Last but not least, in the era of cloud migration, many different cloud providers offer different kinds of price tags. By adopting microservices architecture in design organizations' software systems, one can distribute their systems across cloud services without depending on any of them or even applying a hybrid cloud strategy.

2.3. Cloud Technologies

As the business grows, the increased size of organizations globally has led to IT applications with expanded functionalities, which has escalated operational and IT infrastructure workloads. This propensity has created many challenges to maintain the whole IT infrastructure, especially for small and medium-sized businesses. In contrast to the on-premises IT infrastructure, cloud computing promises to reduce the traditionally high IT infrastructure costs and provide flexible scalability. As a result, companies only need to focus their resources on their core business.

The year 2006 was marked by the first launch of a general-purpose cloud service owned by the public cloud service provider Amazon Web Services (AWS) (Kratzke and Quint, 2017). Since then, the global cloud market is expected to witness exponential growth and will be almost $500 billion by 2022 (Rimol, 2021). In addition, the impact of the COVID-19 pandemic on the economy and society will continue to catalyze digital transformation, innovation, and adoption of cloud services. The remote, collaboration work, and new digital services to enable a hybrid workforce is the examples of mass adoption of cloud service.

Cloud computing is defined as a distributed system, and it consists of a series of interconnected and highly virtualized computers (Fang and Yin, 2010). Through elastic, virtualized infrastructures in which a technology reuses hardware equipment to provide an expandable system environment with extra flexibility, it is possible to lease and release the needed resources in an on-demand, utility-like fashion, with billing according to usage to scale the computing infrastructure up and down rapidly. By adopting cloud hosting, things have become more flexible (Fang and Yin, 2010).

The cloud-native emerged from cloud computing is a series of technologies that empower organizations to build and run applications that take full advantage of the cloud computing model. In general, the term cloud-native applications refers to the applications designed for the cloud. Indeed, containers, service meshes, microservices, immutable infrastructure, and declarative APIs exemplify this design approach. When the business grows, the demand for expanded services to serve more customers and satisfy more business needs becomes higher. The cloud-native applications are designed to embrace rapid change, large scale, and resilience (Cloud Native Computing Foundation, 2015).

Microservices architecture, which has the nature of distributed computing and storage, best suits cloud computing. One can fully deploy microservices applications into public, private, or even hybrid cloud. Microservices also reduce the scope of deployment for applications by enabling the components of the system to be moved around, e.g., public cloud platform or hybrid (i.e., cloud and on-premise).

3. Literature Review on Business Process Compliance Management

A structured literature review of existing studies that contribute to business process compliance is reported in this section. There are two main phases in the review process: literature selection; and literature analysis and evaluation.

We performed the literature selection based on two systematic review articles related to business process compliance (Hashmi and Governatori, 2018; Mustapha *et al.*, 2020). Combined with the recent research article (Ha *et al.*, 2021), it yields 15 selected papers under consideration.

Prior to analyzing the selected papers, we categorized them into three groups based on their topics.

- *Traditional process compliance based on the business environment* focuses on the traditional approach to building the tools or frameworks. Temporal logic, deontic and defeasible logic, Petri nets, and workflow graphs are primarily used the formal language to analyze the compliance of business processes (Awad *et al.*, 2011; Goedertier and Vanthienen, 2006; Governatori and Rotolo, 2010; Ha *et al.*, 2021; Ly *et al.*, 2011; Pesic and van der Aalst, 2006).

- *Cloud SOA-based process compliance* focuses on proposing the tools or framework designed to match the SOA pattern, which eventually can be deployed to the cloud environment. Typical examples are the compliance analysis techniques based on temporal/ deontic logic (Elgammal *et al.*, 2011, 2012), business vocabulary, and business rules/ rule interchange format (Weigand *et al.*, 2011), first-order-logic/ finite state machines (Rodríguez *et al.*, 2013).
- *General cloud-based process compliance* focuses on compliance checking on the general client-server architecture. Except for the work from (Accorsi *et al.*, 2011) utilized Petri nets, the rest of the papers in this category do not specify their formal language to analyze the compliance among business processes (Compagna *et al.*, 2013; Elgammal and Turetken, 2015; Singh and Sidhu, 2017).

Next, we summarized the main contributions from the selected literature in the business process compliance framework and tool.

Table 1 presents a detailed summary of these key papers.

Table 1. Summary of studies in business process compliance

Paper	Frameworks/ Tools	Formal Language	Business Environ-ment	Main Contribution
(Goedertier and Vanthienen, 2006)	PENELOPE	Temporal logic	Traditional	Introduce the PENELOPE language to verify and validate such a set of deontic assignments.
(Pesic and van der Aalst, 2006)	DECLARE	Temporal logic	Traditional	Propose the ConDec language for modeling and enacting dynamic business processes
(Ghose and Koliadis, 2007)	Business Process Compliance Auditing	Computation Tree Logic	Traditional	Propose an auditing framework that utilizes a heuristic approach to verify the compliant status of processes.
(Awad *et al.*, 2011)	BPMN-Q	Computation Tree Logic/ Petri nets	Traditional	Propose a query-based visual language for structural compliance checking of business processes.
(Governatori and Rotolo, 2010)	PCL	Deontic and Defeasible logic	Traditional	Introduce PCL to provide strong conceptual foundations to pride reasoning and modeling support for legal norms.
(Accorsi *et al.*, 2011)	COMCERT	Petri nets	Cloud	Comcert classifies compliance rules from regulatory frameworks for cloud-based compliant workflows. These rules are then formalized in Petri nets for automated detection of non-compliant behavior
(Elgammal *et al.*, 2011)	COMPAS	Temporal/ Deontic Logic	Cloud (SOA)	COMPAS aims to provide all-around compliance governance and can verify compliance of business processes across their lifecycle.
(Ly *et al.*, 2011)	SEAFLOWS	First-order-logic	Traditional	SEAFLOWS is a compliance-by-design framework and a structural compliance checking approach to verify the compliance rules. It can check compliance with behavioral and structural compliance rules.

(Weigand *et al.*, 2011)	Business Policy Compliance in SOA	Business Vocabulary and Business Rules/ Rule Interchange Format	Cloud (SOA)	Provide a formal characterization of behavioral rules for business policy compliance for SOA, which is particularly useful for checking the structural compliance of business processes
(Elgammal *et al.*, 2012)	Tool-suite: CRM/ WSAT/ DCVM	Temporal Logic	Cloud (SOA)	Provide a taxonomy of high-level pattern-based compliance constraints for business processes. Then the analysis is conducted to resolve design-time compliance violations.
(Rodríguez *et al.*, 2013)	SOA-enabled Compliance Management	First-order-logic/ Finite state machines	Cloud (SOA)	Proposed an SOA-enabled compliance management framework for auditing the compliance of SOA based processes
(Compagna *et al.*, 2013)	Security Validation as a Service (SVaaS)	-	Cloud	SVaaS is a means for validating compliance of business processes at the design stage. It provides graphical representation and validation of security in business logic to handle serverside
(Elgammal and Turetken, 2015)	Compliance-Related Ontologies as a Service (COaaS)	-	Cloud	Propose a framework for managing compliance using ontology. The main focus is the compliance repository. It allows for backup of the entire business process compliance lifecycle
(Singh and Sidhu, 2017)	Compliance-based Multi-dimensional Trust Evaluation System (CMTES)	-	Cloud	Compliance-based trust evaluation system to determine the trustworthiness of cloud service providers (CSP). CMTES enables cloud clients to assess the trustworthiness of a CSP from different perspectives, as trust is a subjective concept
(Ha *et al.*, 2021)	Behavioral Relations Compliance-based Architecture (BRECA)	Workflow Graph	Traditional	BRECA is a compliance checking architecture meant to explore the behavioral relations of business processes and the normative requirements. It utilizes the new computation approach (PBR) (Ha and Prinz, 2021) to achieve computational efficiency, resolving most of the BPC framework's main obstacles.

The literature analysis results in Table 1 justify our assertion of the need for new and modern cloud-based business process compliance. As we can see, the traditional business environment has been considered more than the cloud environment (Mustapha *et al.*, 2020). The summary of research contributions revealed that the approaches had been more formal techniques than model checking and semantics. This evidence shows that the research trend on business process compliance centered on the cloud environment is quite active. In addition, the SOA is the most dominant design pattern for a cloud-based environment. There is a considerable amount of approaches focusing on compliance management that relates to SOA architecture.

The drawback of SOA architecture is well-discussed in the previous sections. To the best of the authors' knowledge, none of the existing methods for business process compliance are designed for microservice architecture. Insights gained from the reported evaluation hint at the research gap. The following section addresses the research gap in integrating the microservice into the business process compliance framework (Ha *et al.*, 2021).

4. BPC4MSA: A Modern Cloud-based Microservice Architecture for Business Process Compliance

One of the significant challenges regarding implementing microservice architecture is the decomposition of the system into adequately tailored services. Therefore several different strategies and approaches are available. One possibility to decompose an application is according to business capabilities or use cases (Alpers *et al*., 2015). This approach decomposes the application vertically into a cohesive subsystem called verticals (Alpers *et al.*, 2015).

4.1. Overview of the proposed BPC architecture

Within this section, the paper proposes a microservice-based architecture (MSA) for the business process compliance framework mentioned in Section 2. Since the proposed architecture focuses on BPC, we named this architecture BPC4MSA. As in Fig. 2, the traditional BPC application will be decomposed according to functions and resources based on the mentioned decomposition strategy. A service will cover all independent operations regarding a specific resource (e.g., a Business process model from PAIS space, encoded compliance rules from compliance space). Fig. 4 shows our realization of the conceptual BPC4MSA with a mapping of primary technology choices based on the BPC framework proposed by (Ha *et al.*, 2021; Ha and Prinz, 2021). A summary is provided as follows:

All services are grouped according to three clusters of services types. These types comprise:

- *BPC automation services*: This cluster includes a collection of services that will be the main workhorse for compliance checking of business processes.
- *System management services*: It contains two management services. The first one serves the purpose of monitoring and logging system to track error reporting and related data centrally. The second is utilized to ensure that only authenticated users can access the specific functionalities for which they are authorized.
- *External Integration Services*: The integration services will play a middle man to communicate with external systems such as PAIS to retrieve process models and related metadata and regulation documents parsing tools for machine-readable legal documents.

In addition, the whole architecture is divided into two sides, the client-side and server-side. Although it is possible to call each service directly, the architecture proposed in this paper additionally comprises an API gateway for lightweight message passing. All of them are communicated via API gateways, which support both high-performance communication gRPC (Google Remote Procedure Call) and the standard RESTful API.

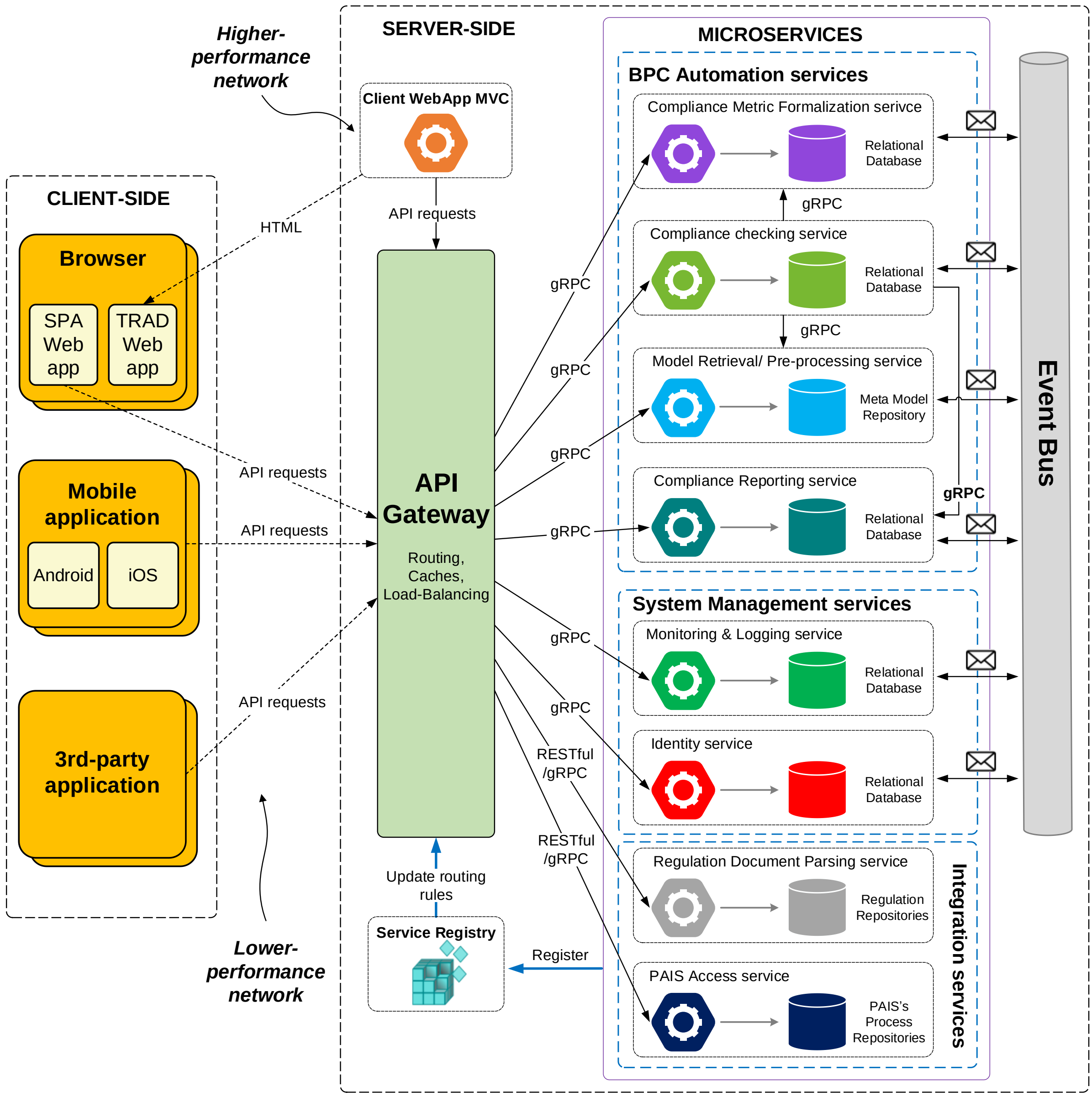

Fig. 4. Realization of the BPC4MSA

As shown in Fig. 4, BPC4MSA offer features covering a broad range of functionalities of traditional BPC, as shown in Fig. 2 (Ha *et al.*, 2021; Ha and Prinz, 2021). To fulfill the requirements of microservice architecture, each service of the business process compliance application (see Fig. 4) runs decentralized in a separate process accessing an individual data storage. This design results in loose coupling and allows the use of a technology stack that is best suited for a particular service. In terms of development, each team that develops a service may choose its technology stack (e.g., C#, Java, Python, JavaScript, or PHP), which best fits its needs.

For simplicity, not all available services are displayed in the diagram. Additional services and types might be added to extend or create tool support for specific scenarios without harming the overall BPC4MSA. It could be seen as the advantage of the microservices design pattern, where we can add more functionalities to the systems independently and quickly.

BPC4MSA enables the user to analyze and simulate the business process's characteristics based on various properties. Analysis results can be aggregated by reporting services that provide the foundation for dashboards functionality. It is worth noticing that all of those services do not require to be hosted in the same physical server or the same cloud service provider.

4.2. API gateway

The API gateway plays an essential role in coordination and managing the flow of information between services and stakeholders. API gateway, a middleware sitting between an API endpoint and backend services, provides lightweight communication technologies, enabling microservices to communicate with others across the physical infrastructure. API Gateway also is incorporated with load balancing, caching, and routing modules to manage various gateways instances dynamically according to the number of user requests. Further cache module can also be added for increasing the performance, as illustrated in Fig. 4.

The client-server communication is implemented using RESTful API and communicates via an API gateway. The API gateway helps reduce the number of (concurrent) service calls and thereby increases the performance (Alpers *et al.*, 2015). However, in contrast to complex ESBs of SOA, the API gateway does not provide any sophisticated routing logic.

4.3. Service Discovery

In the dynamic hosting environment of microservice, in which each service could have a different IP address, we can not statically configure a client with the IP addresses of the services. Instead, the system must use a dynamic service discovery mechanism. Service discovery is conceptually quite simple: its vital component is a *Service Registry*, a database of the network locations of a system's service instances (Richardson, 2018). The service discovery mechanism updates the service registry when service instances start and stop. When a client invokes a service, the service discovery mechanism queries the service registry to obtain a list of available service instances and routes the request to one of them. Service Registry will act as a registration system for microservices and can be consulted by API Gateways and microservices for services discovery. This registry will serve as a repository containing the metadata of all microservices.

4.4. Client-side applications

The client-side applications primarily communicate with the microservices from the server-side via an API gateway. The client-server design pattern enforces the separation of concerns, which helps the client and the server components evolve independently. The communication between client applications and microservices from servers is established via RESTful API. REpresentational State Transfer (REST) is a resource-oriented architecture style presented in 2000 by Roy Fielding's famous dissertation for building internet-scale distributed applications (Fielding, 2000). Meanwhile, RESTful API refers to an Application Programming Interface (API) that obeys the REST architecture style.

The client-side applications request the resources from the server mainly by the RESTful APIs. The endpoint of these APIs, which is one end of a communication channel (*URL*), is aggregated at the API gateway (as shown in Fig. 4). Since the server exposes its functionalities via the APIs, in the proposed BPC4MSA, the types of client applications include, which could be particularly relevant to the BPC context, but are not limited to:

- *Web application*: There are two general approaches to building web applications, the Traditional (TRAD) web app and the Single-Page Application (SPA) web app (Microsoft, 2021). While TRAD web app performs most of the application logic on the server, the SPA performs most of the user interface logic in a web browser, communication with the server primarily using the RESTful APIs. Traditional web applications handle HTTP requests from browsers, get HTML pages from the Client WebApp MVC service, and often provide simple and read-only client apps. Many modern technologies could implement SPA web apps, such as JavaScript, TypeScript, or Blazor WebAssembly (Microsoft, 2021).

- *Mobile application*: It uses the API explicitly created for mobile applications. The need for a mobile app is different from a web app. The APIs might need to optimize even further so that data responses can be more efficient. The mobile app needs to communicate with the API gateway to request the aggregated data from multiple microservices behind the API gateway and return a single set of data, possibly eliminating any data in the response that is not needed by the mobile app. The returning data might be compressed to reduce bandwidth consumption. Therefore, it decreases network traffic and improves the server's response times.
- *Third-party application*: The server also exposes the public APIs implemented for third-party developers. The third-party applications can access the public APIs over the internet, so API composition is likely inefficient (Richardson, 2018). But the inefficiency of API composition is a relatively minor problem compared to the much more significant challenge of designing an API that each third-party application uses. That is because third-party developers need a stable API. In general, the APIs for a third-party application often have a general purpose that is not optimized for each use case, like mobile or web applications.

4.5. BPC Automation services

Business process compliance automation services are the core of BPC4MSA. It includes four primary services that ensure the checking of compliance functionality works correctly. The communication between these services is established via a high-performance gRPC protocol. Each service also reports its state or related event to the event bus so that other services can get notified when needed. The detailed functionality of four microservices in this cluster is as follows:

- *Compliance Metric Formalization service*: It retrieves the compliance rules in the machine-readable format from the Regulation document parsing service via the API gateway. Then these rules are converted into the compliance metrics, the *Behavioral Relations* (BhR) as described in (Ha *et al.*, 2021), and ready to feed to the compliance checking service
- *Model retrieval/ pre-processing service*: It retrieves the process models and their related metadata from the PAIS access service via the API gateway. Then the models and their related information are pre-processed if needed and ready to be used.
- *Compliance checking service*: This service is empowered by the cutting edge computational method, the *Partitioning Behavioral Retrieval* (PBR), proposed by (Ha and Prinz, 2021). All processing power of the BPC4MSA will focus on this service as it will be the one that takes the heaviest workload. Compliance checking service gets input BhR from the *Compliance metric Formalization service* and the information of business process models from the *Model retrieval/pre-processing service*. Then it produces the inference about the compliance level of the given process model based on the required compliance rules encoded in BhR. The communication

is also facilitated by gRPC via direct communication between services or via API gateway. The output of this service can then be forwarded to the *Compliance reporting service*.

- *Compliance reporting service*: It creates and exports reports based on the clients' demands. It also provides visualization templates representing different findings and insights from the given input. The input of this service is retrieved from the *Compliance checking service* via gRPC.

4.6. System management services

Two primary services allow for managing the whole system. Firstly, the *identity service* enables the server to implement the much-needed protection and provide access to business services spanning many microservices only to authorized clients. This service may include implementing message encryptions, access tokens, or transactions for authorization and authentication. It includes client and system authentications, authentications between microservices, and third-party authentications. Secondly, the *monitoring and logging service* allows BPC4MSA to check the health of microservices and the whole system's performance. This service includes an active check on the microservices and the running operations. It also provides an analytics functionality that shows microservices' runtime statistics and operations.

4.7. Event Bus

The Event Bus (or message channel/bus) employed in this architecture enables the microservices to have the publish/subscribe-style communication without explicitly requiring the components to be aware of each other. We can also describe the format of the exchanged messages using a standard such as JSON, XML, or Protobuf. A service can publish events using a publish/subscribe interaction style. The event and topics model of messaging is a great abstraction and an excellent way to design a service’s asynchronous API. But to implement a service, we need to choose an event bus technology and determine how to implement your design using its capabilities. A messaging-based application typically uses a message broker, an infrastructure service through which the service communicates (Richardson, 2018). A message broker is an intermediary through which all messages flow.

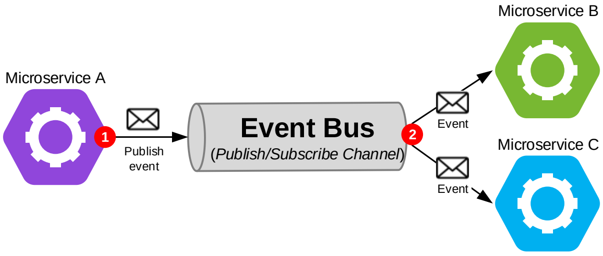

Fig. 5. Publish/subscribe basis with an event bus

As shown in Fig. 5, microservice A publishes to Event Bus, which distributes to subscribed microservices B and C without the publisher knowing the subscribers. The event bus is related to the *Observer* pattern, in which the primary object notifies other interesting objects with relevant information (event). There are multiple potential implementations for event buses, using a different technology or infrastructure such as RabbitMQ, Azure Service Bus, Google Cloud Pub/Sub, or any other third-party open-source or commercial.

5. Conclusion

Regulation compliance has been studied in other fields, e.g., law and legal reasoning, but less attention has been paid to business process management and cloud-based microservices architecture. Cloud migration has become an irreversible trend for organizations regarding digital transformation. Microservices architecture is recent inception and is being adopted at an increased pace by the industry. It certainly offers advantages over various concerns of previous architectures, from monolithic to Service-Oriented Architecture (SOA). Since business process compliance frameworks and tools enable businesses to be subject to strict regulations for effective, transparent operations and a dynamic business environment, a holistic review of the available literature on business process compliance frameworks and tools is also presented. Most of the approaches focused on proposing the compliance framework for traditional business environments, while others focused on the cloud-based SOA. As a result, it indicates the need for a BPC framework that best suits the microservice design pattern in a cloud-based business environment.

2020 has been a year of digital transformation and acceleration to meet the challenges of the COVID-19 pandemic (Thomson Reuters, 2021). Companies have swiftly implemented the roll-out of resilience technology to allow business activities to carry on as nations went into lockdown. As a core pillar of digital transformation, FinTech may take a light-year leap post-COVID 19. Regtech, on the other hand, is defined as a subset of fintech focusing on regulatory compliance technologies more efficiently and effectively. As fintech grows, regulators are responding to the advancement of fintech by encouraging and employing regulation technologies such as BPC to enforce compliance in companies' business processes (Thomson Reuters, 2021). Therefore, developing a resilience BPC solution as a regtech service could bring huge business interest for companies rapidly seizing the opportunity in the new normal post-COVID-19 era.

BPC4MSA is the first resilience framework to provide a fully functional BPC empowered by the microservices architecture. It is designed based on the most recent and prominent traditional BPC framework proposed by (Ha *et al.*, 2021). Despite the advantages of employing microservice architecture, some issues need to be addressed. For example, decomposing monolithic applications into separated services and data management across independent database silos is not trivial. However, the mass adoption of microservice architecture indicates that using microservice outweighs the disadvantages.

For future research opportunities, the work presented here can be extended further. First, we plan to build a proof of the concept of a business process compliance system based on BPC4MSA to examine how the system can be used in actual practice. Further on, a rigorous empirical study comparing different contemporary BPC systems could be conducted to provide hints to researchers and practitioners in the context of business process compliance in a cloud-based business environment.

References

Accorsi, R., Lowis, L. and Sato, Y. (2011), "Automated Certification for Compliant Cloud-based Business Processes", *Business & Information Systems Engineering*, Vol. 3 No. 3, pp. 145–154.

Alpers, S., Becker, C., Oberweis, A. and Schuster, T. (2015), "Microservice Based Tool Support for Business Process Modelling", *2015 IEEE 19th International Enterprise Distributed Object Computing Workshop*, IEEE, pp. 71–78.

Awad, A., Weidlich, M. and Weske, M. (2011), "Visually specifying compliance rules and explaining their violations for business processes", *Journal of Visual Languages and Computing*, Elsevier, Vol. 22 No. 1, pp. 30–55.

Binh, L.D. and Phuong, T.T. (2020), *Digital Economy And Digital Transformation in Vietnam*, Hanoi.

Cameron, Pham, T.H., Atherton, J., Nguyen, D.H., Nguyen, T.P., Nguyen, T.N., Trinh, H.Y., *et al.* (2019), *Vietnam's Future Digital Economy - Towards 2030 and 2045*, CSIRO, Brisbane.

Cloud Native Computing Foundation. (2015), "Mission of the Cloud Native Computing Foundation", available at: https://github.com/cncf/foundation/blob/main/charter.md (accessed 21 March 2022).

Compagna, L., Guilleminot, P. and Brucker, A.D. (2013), "Business process compliance via security validation as a service", *Proceedings - IEEE 6th International Conference on Software Testing, Verification and Validation, ICST 2013*, pp. 455–462.

Dragoni, N., Giallorenzo, S., Lafuente, A.L., Mazzara, M., Montesi, F., Mustafin, R. and Safina, L. (2017), "Microservices: Yesterday, Today, and Tomorrow", *Present and Ulterior Software Engineering*, Springer International Publishing, Cham, pp. 195–216.

Dumas, M., La Rosa, M., Mendling, J. and Reijers, H.A. (2013), *Fundamentals of Business Process Management*, Springer Berlin Heidelberg, Berlin, Heidelberg, available at:https://doi.org/10.1007/978-3-642-33143-5.

Elgammal, A. and Turetken, O. (2015), "Lifecycle Business Process Compliance Management: A Semantically-Enabled Framework", *2015 International Conference on Cloud Computing (ICCC)*, IEEE, pp. 1–8.

Elgammal, A., Turetken, O. and Van Den Heuvel, W.-J. (2012), "Using Patterns for the Analysis and Resolution of Compliance Violations", *International Journal of Cooperative Information Systems*, Vol. 21 No. 01, pp. 31–54.

Elgammal, A., Turetken, O., van den Heuvel, W.-J. and Papazoglou, M. (2011), "On the Formal Specification of Regulatory Compliance: A Comparative Analysis", in Maximilien, E.M., Rossi, G., Yuan, S.-T., Ludwig, H. and Fantinato, M. (Eds.), *Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, Vol. 6568, Springer Berlin Heidelberg, Berlin, Heidelberg, pp. 27–38.

Fang, Z. and Yin, C. (2010), "BPM Architecture Design Based on Cloud Computing", *Intelligent Information Management*, Vol. 02 No. 05, pp. 329–333.

Fielding, R.T. (2000), *Architectural Styles and the Design of Network-Based Software Architectures*, University of California, Irvine.

Ghose, A. and Koliadis, G. (2007), "Auditing Business Process Compliance", *Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, Vol. 4749 LNCS, pp. 169–180.

Goedertier, S. and Vanthienen, J. (2006), "Designing Compliant Business Processes with Obligations and Permissions", *Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, Vol. 4103 LNCS, pp. 5–14.

Governatori, G. (2005), "Representing Business Contracts In RuleML", *International Journal of Cooperative Information Systems*, Vol. 14 No. 02n03, pp. 181–216.

Governatori, G. and Rotolo, A. (2010), "A conceptually rich model of business process compliance", *Conferences in Research and Practice in Information Technology Series*, Vol. 110, pp. 3–12.

Governatori, G. and Sadiq, S. (2008), "The Journey to Business Process Compliance", *Handbook of Research on Business Process Modeling*, pp. 426–454.

Ha, N.L., Nguyen, T.T. and Do, S.H. (2021), "Application of Business Process Compliance Technology to the Digital Transformation of Financial Industries from Fintech and Regtech Perspective", *Tạp Chí Khoa Học Quản Lý Và Kinh Tế, Trường Đại Học Kinh Tế, Đại Học Huế*, Vol. 18, available at: https://tapchi.hce.edu.vn/index.php/sjme/article/view/73.

Ha, N.L. and Prinz, T.M. (2021), "Partitioning Behavioral Retrieval: An Efficient Computational Approach With Transitive Rules", *IEEE Access*, IEEE, Vol. 9, pp. 112043–112056.

Hashmi, M. and Governatori, G. (2018), “Norms modeling constructs of business process compliance management frameworks: a conceptual evaluation”, *Artificial Intelligence and Law*, Springer Netherlands, Vol. 26 No. 3, pp. 251–305.

Hashmi, M., Governatori, G., Lam, H.P. and Wynn, M.T. (2018), “Are we done with business process compliance: state of the art and challenges ahead”, *Knowledge and Information Systems*, Springer London, Vol. 57 No. 1, pp. 79–133.

Hill, J.B., Sinur, J., Flint, D. and Melenovsky, M.J. (2006), *Gartner’s Position on Business Process Management*, *Gartner Research G*, Vol. 136533.

Kharbili, M. el, Medeiros, A.K.A. de, Stein, S. and Aalst, W.M.P. van der. (2008), “Business process compliance checking: Current state and future challenges”, in Loos, P., Nüttgens, M., Turowski, K. and Werth, D. (Eds.), *Modellierung Betrieblicher Informationssysteme (MobIS 2008)*, Gesellschaft für Informatik e.V., Bonn, pp. 107–113.

Kratzke, N. and Quint, P.C. (2017), “Understanding cloud-native applications after 10 years of cloud computing - A systematic mapping study”, *Journal of Systems and Software*, Elsevier Inc., Vol. 126, pp. 1–16.

López, H.A., Debois, S., Slaats, T. and Hildebrandt, T.T. (2020), “Business process compliance using reference models of law”, *Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, Vol. 12076 LNCS, pp. 378–399.

Ly, L.T., Knuplesch, D., Rinderle-Ma, S., Göser, K., Pfeifer, H., Reichert, M. and Dadam, P. (2011), “SeaFlows Toolset – Compliance Verification Made Easy for Process-Aware Information Systems”, *Lecture Notes in Business Information Processing*, Vol. 72 LNBIP, pp. 76–91.

Messina, A., Rizzo, R., Storniolo, P. and Urso, A. (2016), “A Simplified Database Pattern for the Microservice Architecture”, *The Eighth International Conference on Advances in Databases, Knowledge, and Data Applications*, No. June, pp. 35–40.

Microsoft. (2021), “Choose Between Traditional Web Apps and Single Page Apps (SPAs)”, *Microsoft Docs*, available at: https://docs.microsoft.com/en-us/dotnet/architecture/modern-web-apps-azure/choose-between-traditional-web-and-single-page-apps (accessed 24 March 2022).

Møller, C., Maack, C.J. and Tan, R.D. (2007), “What is Business Process Management: A Two Stage Literature Review of an Emerging Field”, *Research and Practical Issues of Enterprise Information Systems II Volume 1*, Springer US, Boston, MA, pp. 19–31.

Mustapha, A.M., Arogundade, O.T., Misra, S., Damasevicius, R. and Maskeliunas, R. (2020), “A systematic literature review on compliance requirements management of business processes”, *International Journal of Systems Assurance Engineering and Management*, Springer India, Vol. 11 No. 3, pp. 561–576.

Pesic, M. and van der Aalst, W.M.P. (2006), “A Declarative Approach for Flexible Business Processes Management”, *Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, Vol. 4103 LNCS, pp. 169–180.

Richardson, C. (2018), *Microservices Patterns: With Examples in Java*, Simon and Schuster, available at: https://www.manning.com/books/microservices-patterns.

Rimol, M. (2021), “Four Trends Are Shaping the Future of Public Cloud”, *Gartner*, available at: gartner.com/en/newsroom/press-releases/2021-08-02-gartner-says-four-trends-are-shaping-the-future-of-public-cloud (accessed 21 March 2022).

Rodríguez, C., Schleicher, D., Daniel, F., Casati, F., Leymann, F. and Wagner, S. (2013), “SOA-enabled compliance management: Instrumenting, assessing, and analyzing service-based business processes”, *Service Oriented Computing and Applications*, Vol. 7 No. 4, pp. 275–292.

Singh, S. and Sidhu, J. (2017), “Compliance-based Multi-dimensional Trust Evaluation System for determining trustworthiness of Cloud Service Providers”, *Future Generation Computer Systems*, Elsevier B.V., Vol. 67, pp. 109–132.

Thomson Reuters. (2021), *FinTech, RegTech and The Role Of Compliance 2021*.

Weigand, H., van den Heuvel, W.-J. and Hiel, M. (2011), “Business policy compliance in service-oriented systems”, *Information Systems*, Elsevier, Vol. 36 No. 4, pp. 791–807.