

# ĐỀ XUẤT MÔ HÌNH QUẢN TRỊ TUÂN THỦ QUY TRÌNH DỰA TRÊN NỀN TẢNG ĐIỆN TOÁN Đám Mây

## INTEGRATION OF BUSINESS PROCESS COMPLIANCE FRAMEWORK AND MODERN CLOUD-BASED ARCHITECTURE

Hà Ngọc Long\*, Đỗ Sông Hương, Nguyễn Hoàng

Trường Đại học Kinh tế, Đại học Huế<sup>1</sup>

\*Tác giả liên hệ: hnlong@hueuni.edu.vn

(Nhận bài: 02/10/2022; Chấp nhận đăng: 16/12/2022)

**Tóm tắt** - Các giải pháp phần mềm quản lý quy trình nghiệp vụ đã được sử dụng nhằm góp phần cải thiện năng lực hoạt động của tổ chức và doanh nghiệp. Do đó, nhu cầu kiểm tra tuân thủ (BPC) trong các quy trình nghiệp vụ ngày càng trở nên cấp thiết. Microservices (MSA) nổi lên như là một kiến trúc tiềm năng cho lĩnh vực điện toán đám mây. Việc chuyển đổi từ các hệ thống xây dựng truyền thống sang nền tảng đám mây, đặc biệt trên kiến trúc microservice, đã và đang là xu thế tất yếu trong giai đoạn chuyển đổi số và cách mạng công nghiệp 4.0 hiện nay. Bài báo này trước tiên sẽ tổng hợp các nghiên cứu liên quan đến lĩnh vực tuân thủ đối với các quy trình nghiệp vụ. Dựa trên những lỗ hổng nghiên cứu được chỉ ra, bài báo này đề xuất một mô hình tuân thủ quy định được thiết kế dành riêng cho microservices, với tên gọi là BPC4MSA. BPC4MSA sử dụng kỹ thuật tính toán mới trong lĩnh vực nghiên cứu tuân thủ quy trình nghiệp vụ.

**Từ khóa** - Điện toán đám mây; hệ thống thông tin; microservices; quản lý quy trình nghiệp vụ; tuân thủ quy trình nghiệp vụ

### 1. Đặt vấn đề

Việt Nam gần đây đã phê duyệt *Chương trình quốc gia về chuyển đổi số đến năm 2025, tầm nhìn đến năm 2030*, nhằm đẩy nhanh quá trình chuyển đổi số thông qua những thay đổi về nhận thức, chiến lược doanh nghiệp và khuyến khích hướng tới số hóa doanh nghiệp, hoạt động quản trị và sản xuất. *Chuyển đổi số* (CDS) là một quá trình các tổ chức, doanh nghiệp (mà sau đây gọi chung là *tổ chức*) sử dụng công nghệ để tự động hóa và tối ưu hóa quy trình nghiệp vụ và kinh doanh của họ cũng như nâng cao trải nghiệm của khách hàng. Kết quả là, các tổ chức luôn có sức cạnh tranh và thích nghi tốt, từ đó giúp đạt được các mục tiêu phát triển [1].

Trong môi trường kinh doanh, nghiệp vụ theo định hướng quy trình, các tổ chức tập trung vào việc tạo lập, tối ưu hóa và giám sát các quy trình kinh doanh, nghiệp vụ của họ, từ đó giúp giảm chi phí, tăng doanh thu và cải thiện hiệu quả hoạt động. Lĩnh vực *Quản trị Quy trình Nghiệp vụ* (Business Process Management) đã nổi lên như một công nghệ tổng thể nhằm hỗ trợ việc kiểm soát và vận hành toàn bộ hoạt động kinh doanh thông qua các quy trình rõ ràng, từ đó giúp quản lý các quy trình này một cách có hệ thống. BPM giúp tạo ra lợi thế cạnh tranh cho tổ chức bằng cách cải thiện sự linh hoạt của tổ chức trong dài hạn [2]. Lợi ích chính của BPM là khả năng điều chỉnh các quy trình nghiệp vụ phù hợp với các yêu cầu thay đổi của thị trường, cho phép các doanh nghiệp phản ứng nhanh hơn và hiệu quả hơn với sự biến động liên tục của thị trường [3]. Nhiều

**Abstract** - Business process management software solutions have been used to improve the operational capacity of organizations and businesses. Therefore, the need for compliance checking in business processes is becoming more and more urgent. Microservices (MSA) is emerging as a potential architecture for the cloud computing field. The transition from on-premises systems to the cloud platform, especially in microservices architecture, has been an inevitable trend in digital transformation and the Industrial Revolution 4.0. This article will first summarize the research related to the field of compliance for business processes (BPC). Based on the structural literature review, this paper proposes a regulatory compliance framework designed explicitly for the microservice architecture, named BPC4MSA, that employs cutting-edge technologies in business process compliance research.

**Key words** - Business process management; Business process compliance; Cloud computing; Microservices; Information Systems

tổ chức đã tích hợp các giải pháp phần mềm BPM vào cơ sở hạ tầng công nghệ thông tin (CNTT) của họ. Các giải pháp BPM (BPM Solution - BPMS) phổ biến thường được triển khai trên môi trường đám mây, mang lại sự linh hoạt hơn và giảm chi phí đầu tư ban đầu cho tổ chức.

Tuy nhiên, các quy trình nghiệp vụ luôn phải tuân theo các quy định của luật pháp, chuẩn mực trong nước hoặc quốc tế. Do đó, việc đảm bảo các quy trình kinh doanh của tổ chức tuân thủ các quy định, chuẩn mực và luật pháp là điều cần thiết đối với bất kỳ BPMS nào [4]. Thuật ngữ *Tuân thủ Quy trình Nghiệp vụ* (Business Process Compliance) được sử dụng để mô tả sự phù hợp của các quy trình nghiệp vụ với các quy định của pháp luật, các chuẩn mực và hướng dẫn có liên quan [5]. Trong phạm vi của nghiên cứu này, sẽ tập trung vào việc kiểm tra tuân thủ quy trình kinh doanh ở ngay từ giai đoạn thiết kế quy trình.

Một trong những trụ cột chính của chuyển đổi số phải kể đến là Công nghệ điện toán đám mây (Cloud computing). Làn sóng tiếp theo của công nghệ kỹ thuật số - điện toán đám mây cùng với trí tuệ nhân tạo (AI), công nghệ chuỗi khối (blockchain), internet vạn vật (IoT) và các dịch vụ dựa trên đám mây - có tiềm năng đưa Việt Nam thành nền kinh tế có tốc độ phát triển cao và nâng cao mức sống của người dân Việt Nam trong những thập kỷ tới [6].

Các nghiên cứu gần đây cho thấy, việc ưu tiên sử dụng các phần mềm được cài đặt tại chỗ (on-premise) đang được coi trọng hơn so với môi trường đám mây (cloud) [7].

<sup>1</sup> University of Economics, Hue University (N. Long Ha, S. Huong Do, Hoang Nguyen)

Ngoài ra, hầu hết các phương pháp BPC dựa trên đám mây đều lựa chọn kiến trúc hướng dịch vụ (Service Oriented Architecture), được coi là một công nghệ công kênh so với kiểu thiết kế mới chú trọng vào sự linh hoạt, như kiến trúc microservices [8]. Microservices là sự phát triển tiếp theo của kiến trúc thiết kế hệ thống phục vụ cho môi trường điện toán đám mây. Nguyên lý chung của microservices đề xuất rằng các hệ thống phần mềm nên được phân tách thành các khối độc lập và nhỏ gọn (microservices), mỗi khối tập trung vào một tác vụ hoặc nhóm tác vụ duy nhất mà nó hoạt động hiệu quả nhất. Các microservices giao tiếp với nhau bằng giao diện lập trình ứng dụng (API) và chúng có thể được cập nhật độc lập và liên tục. Có nhiều khoảng trống nghiên cứu trong chủ đề tuân thủ quy trình nghiệp vụ và điện toán đám mây, đặc biệt là những nghiên cứu dựa trên kiến trúc microservices. Với những dẫn dắt ở trên, đóng góp chính của bài báo có thể kể đến như sau:

- 1) Giới thiệu các khái niệm liên quan đến chủ đề của bài báo, bao gồm: BPM, BPC, microservices và công nghệ đám mây.
- 2) Thực hiện đánh giá tổng quan về các nghiên cứu liên quan đến các phương pháp tiếp cận tuân thủ quy trình nghiệp vụ. Từ đó, chỉ ra những lỗ hổng nghiên cứu cần được hoàn thiện.
- 3) Đề xuất kiến trúc microservice tích hợp vào khung tuân thủ quy trình nghiệp vụ sử dụng kỹ thuật tính toán mới đã được chứng minh tính hiệu quả [9].

## 2. Cơ sở lý thuyết

### 2.1. Quản trị quy trình nghiệp vụ

Trong môi trường kinh doanh định hướng theo quy trình, quy trình chính là xương sống của mỗi một tổ chức để đảm bảo bộ máy hoạt động trơn tru và hiệu quả. Khi hoạt động nghiệp vụ, kinh doanh không ngừng phát triển, nhu cầu về hệ thống thông tin để quản lý các quy trình trở nên cấp thiết hơn bao giờ hết. Kết quả, lý thuyết về quản trị quy trình nghiệp vụ (BPM) đã xuất hiện để hỗ trợ hiệu quả hoạt động và tối ưu hóa quy trình nghiệp vụ cho tổ chức [10].

BPM giám sát cách thức thực hiện công việc trong một tổ chức để đảm bảo rằng các kết quả đạt được là nhất quán, thêm vào đó các cơ hội cải tiến quy trình được ghi nhận một cách có hệ thống [10]. Do đó, ý tưởng trọng tâm của

BPM là tập trung vào việc xác định, thiết kế, tích hợp, tối ưu hóa, thực thi, giám sát và quản lý các quy trình nghiệp vụ.

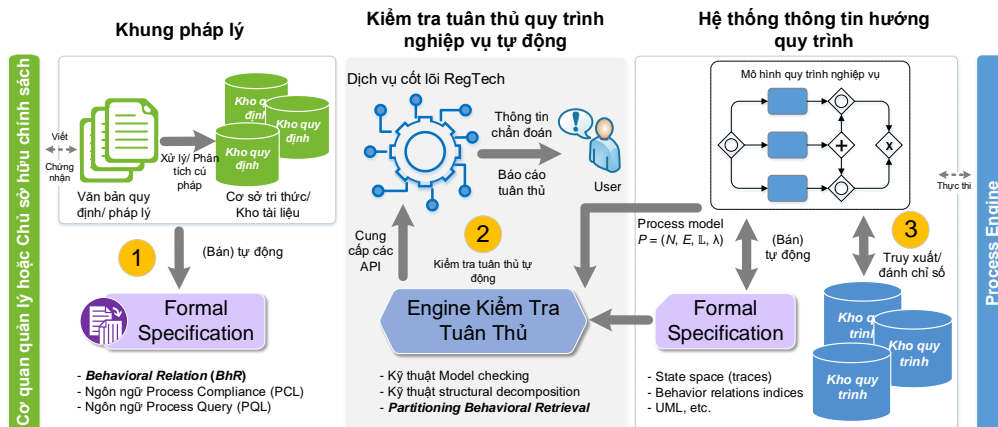
Sự thành công của bất kỳ dự án BPM nào đều cũng đến từ hoạt động mô hình hóa quy trình nghiệp vụ. Nếu không có thiết kế tốt, bất kỳ hệ thống vận hành quy trình nào cũng có thể thất bại [10], [11]. Vì vậy, trong phạm vi của nghiên cứu này, bài báo sẽ tập trung vào các mô hình quy trình nghiệp vụ trong giai đoạn thiết kế (modeling phase).

### 2.2. Tuân thủ quy trình nghiệp vụ

Các quy trình nghiệp vụ luôn phải tuân thủ các luật và quy định hiện hành do các cơ quan quản lý cũng như nội bộ tổ chức áp đặt. Cùng với sự phát triển của kinh tế xã hội, các ràng buộc này ngày càng bị thắt chặt và tăng lên, song song với đó, luật và quy định luôn được cập nhật và bổ sung để điều chỉnh hoạt động sản xuất kinh doanh và quản lý. Đảm bảo tuân thủ quy trình nghiệp vụ là mối quan tâm trọng tâm của bất kỳ tổ chức nào, đặc biệt là sau cuộc khủng hoảng tài chính năm 2008. Tuy nhiên, quá trình kiểm tra tuân thủ thường tốn kém về thời gian và tiền bạc. Do đó, nhiều tổ chức đã chuyển sang sử dụng công nghệ để số hóa quá trình kiểm tra tuân thủ. Một ngành công nghiệp mới, bước ra từ Fintech, được gọi là RegTech (tạm dịch là công nghệ điều tiết) đã được hình thành từ xu thế này. Regtech là sự tổng hợp của các chuyên ngành khác nhau, từ lý thuyết pháp lý đến khoa học máy tính. Sự phổ biến của Regtech được xúc tác bởi sự phát triển của lĩnh vực trí tuệ nhân tạo, dữ liệu lớn, và hệ thống thông tin [12].

Tuân thủ (compliance) là “hành động / quy trình để đảm bảo rằng các hoạt động và quy trình kinh doanh phù hợp với các quy định (thường mang tính pháp lý)” [13]. Trong đó, quy trình nghiệp vụ là nhân tố chính để bắt cứ tổ chức nào hoàn thành các mục tiêu của họ [14]. Chủ đề nghiên cứu về BPC thu hút được sự chú ý ngày càng tăng từ các nhà nghiên cứu và các nhà thực hành cùng với sự phát triển của BPM và RegTech. *Tuân thủ quy trình nghiệp vụ ngay trong giai đoạn thiết kế quy trình* đảm bảo rằng tập hợp các lộ trình thực tế (traces) của một mô hình quy trình mô tả một cách chính xác hành vi thực thi (execution behavior) của nó như đã thiết kế [9]. Do đó, BPC có thể được xem là một phần của RegTech hoặc là sự giao thoa giữa BPM và RegTech.

Khung tuân thủ quy trình nghiệp vụ (BRECA) [15], được mô tả trong Hình 1, diễn tả sự tương tác giữa ba vùng



Hình 1. Khung tuân thủ quy trình nghiệp vụ (BRECA)

(space): (1) Các khung quy định ràng buộc; (2) Tự động hóa tuân thủ quy trình nghiệp vụ; và (3) Hệ thống thông tin nhận thức quy trình (PAIS). Cốt lõi của BPC framework là công cụ kiểm tra tuân thủ. Nó thu thập và tính toán sự tuân thủ dựa trên các thông số kỹ thuật được chuyển đổi từ các quy định từ vùng đầu tiên và mô hình quy trình từ vùng thứ ba. Đặc điểm khác biệt của khuôn khổ này là nó sử dụng một phương pháp mới được gọi là *Partitioning Behavioral Retrieval* (PBR) để đạt được hiệu quả tính toán cao, từ đó tăng tốc quá trình kiểm tra tuân thủ [9].

Trong môi trường kinh doanh thay đổi liên tục, các giải pháp phần mềm BPC được sử dụng trong các tổ chức cần phải nhanh chóng thích ứng với những thay đổi trong: (i) Môi trường kinh doanh (chẳng hạn như nhu cầu của khách hàng) và (ii) những thay đổi liên tục trong quy định, tiêu chuẩn và luật pháp [16]. Do đó, khả năng mở rộng và sự linh hoạt là chìa khóa để một tổ chức đạt được khả năng cạnh tranh. Trong khi khả năng mở rộng đề cập đến việc tăng khả năng phục vụ nhiều khách hàng hơn, chẳng hạn như việc dễ dàng mở rộng cơ sở hạ tầng CNTT, sự linh hoạt đề cập tới cách một hệ thống có thể phản ứng với các thay đổi và được tái cấu trúc, cấu hình lại nhanh chóng và dễ dàng như thế nào. Khung BPC hiện tại (Hình 1) được đề xuất bởi Hà và cộng sự [15] chưa chú trọng về khả năng mở rộng và tính linh hoạt vì mỗi quan tâm chính của nó là làm thế nào để kiểm tra sự tuân thủ của các quy trình kinh doanh một cách hiệu quả. Kiến trúc BPC dựa trên công nghệ đám mây hiện đại nổi lên như một giải pháp tiềm năng để giải quyết các nhược điểm trên. Cách các công nghệ đám mây hiện đại có thể cải tiến khung BPC hiện tại sẽ được thảo luận kỹ hơn ở phần sau của bài báo.

### 2.3. Công nghệ điện toán đám mây và microservices

Khi doanh nghiệp phát triển, quy mô ngày càng được mở rộng, điều đó đã dẫn đến việc các chức năng của hệ thống CNTT cần được mở rộng theo. Xu hướng này đã tạo ra nhiều thách thức để duy trì toàn bộ cơ sở hạ tầng CNTT, đặc biệt là đối với các doanh nghiệp vừa và nhỏ. Trái ngược với cơ sở hạ tầng CNTT được thiết lập tại chỗ (on-premise), điện toán đám mây (cloud computing) hứa hẹn sẽ giảm chi phí đầu tư cơ sở hạ tầng CNTT và cung cấp khả năng mở rộng linh hoạt hơn. Nhờ đó, các công ty chỉ cần tập trung nguồn lực vào hoạt động kinh doanh cốt lõi.

Điện toán đám mây được định nghĩa là một hệ thống phân tán và bao gồm một loạt các máy tính được kết nối với nhau và được ảo hóa cao (highly virtualized) [17]. Thông qua cơ sở hạ tầng ảo hóa, tổ chức có thể cho thuê và giải phóng các tài nguyên cần thiết theo nhu cầu và thanh toán tùy theo mức sử dụng cơ sở hạ tầng CNTT. Bằng cách sử dụng điện toán đám mây, việc thuê mượn hạ tầng CNTT đã trở nên linh hoạt hơn [17].

Ở giai đoạn khởi đầu của kiến trúc phần mềm, các hệ thống phần mềm thường được xây dựng theo *kiểu nguyên khối* (monolithic). Đặc trưng của kiểu xây dựng này là tất cả các thành phần ứng dụng được gom lại vào một khối duy nhất. Ví dụ, giao diện người dùng (UI), logic nghiệp vụ và quyền truy cập dữ liệu (data access), được gom vào một mã nguồn lớn (codebase) duy nhất. Do đó, ứng dụng không được tách rời về mặt bản chất. Điều đó có nghĩa là mỗi mô-đun hoặc thành phần của ứng dụng không thực hiện các nhiệm vụ của nó một cách độc lập. Kiến trúc monolithic là

một phương pháp xây dựng hệ thống theo kiểu truyền thống và trên thực tế, nhiều doanh nghiệp vẫn sử dụng nó.

Ngày nay, nhờ mạng lưới truyền thông có tính khả dụng và tốc độ cao, hầu hết các ứng dụng CNTT đều được xây dựng theo kiểu phân tán. Việc sử dụng ứng dụng nguyên khối trong môi trường phân tán trở nên khó khăn, đặc biệt là khi ứng dụng phát triển ngày càng lớn. Thêm vào đó, trong môi trường kinh doanh thay đổi nhanh chóng, kiến trúc monolithic dường như không thể thích ứng với những thay đổi nhanh chóng do những hạn chế nhất định: (1) Khả năng mở rộng hạn chế; (2) Trở nên ngày càng phức tạp theo thời gian; (3) Khó mở rộng quy mô; (4) Khóa cứng về mặt công nghệ; Và (5) vấn đề về tính khả dụng (availability) [18].



Hình 2. Quá trình tiến hóa của kiến trúc phần mềm

Để nhanh chóng đáp ứng các điều kiện và nhu cầu của thị trường, các ứng dụng của doanh nghiệp cần phải liên tục cập nhật và mở rộng quy mô ở một số thành phần nhưng không được dùng toàn bộ hệ thống. Kiến trúc *microservices* đã được đề xuất để giải quyết các vấn đề nêu trên. *Microservices* đại diện cho sự phân rã các hệ thống monolithic thành các dịch vụ nhỏ hơn (micro-services) và có thể được triển khai một cách độc lập [19]. Trên thực tế, *microservices* là cải tiến tiếp theo của khái niệm kiến trúc hướng dịch vụ (SOA). SOA là thể hệ đầu tiên của các mẫu thiết kế hướng dịch vụ, và thường sử dụng các công nghệ phức tạp và công kênh như SOAP và các tiêu chuẩn WS\* (web service\*) khác [8]. Ngược lại, kiến trúc *Microservices* nhấn mạnh việc tối đa hóa phân tách các hệ thống lớn thành các thành phần độc lập, nhỏ hơn. Không giống như SOA, thường sử dụng ESB để hỗ trợ giao tiếp giữa các dịch vụ, các *microservices* sử dụng *event bus* để giao tiếp [20]. Do đó, quá trình xử lý chỉ diễn ra bên trong mỗi *microservice*, trong khi việc trao đổi thông tin giữa các dịch vụ dựa trên các cơ chế giao tiếp đơn giản (chẳng hạn như REST hoặc *message bus* đơn giản) [20]. Do đó, *microservice* (1) cho phép tùy biến cao hơn và dễ bảo trì; (2) Đảm bảo hệ thống hoạt động liên tục; (3) Cho phép mở rộng độc lập, phân phối và triển khai liên tục (CI/CD) các dịch vụ lớn và phức tạp; (4) Cho phép các nhóm nhà phát triển tự chủ và độc lập về công nghệ sử dụng; (5) Cho phép dễ dàng thử nghiệm và áp dụng các công nghệ mới; (6) Có khả năng phát hiện, cách ly và phục hồi lỗi tốt hơn [8], [18].

Tuy nhiên, kiến trúc này vẫn tồn tại một số vấn đề, chẳng hạn như: (1) Những thách thức về cách chia tách một ứng dụng thành các phần nhỏ hơn một cách có hệ thống; (2) Khó khăn trong việc phát triển, thử nghiệm và triển khai các hệ thống phân tán; (3) Đòi hỏi sự nghiêm ngặt trong việc phối hợp triển khai các tính năng mà trải dài trên nhiều dịch vụ [8].

Trên thực tế, các tổ chức có thể nắm bắt các sáng kiến chuyển đổi số nhanh hơn bằng cách áp dụng *microservices* như một phần của quá trình dịch chuyển đám mây của họ. Các nhà cung cấp dịch vụ đám mây như Microsoft Azure, Google Cloud Platform, Amazon Web Service, v.v. đã và đang là những bên tích cực trong việc hỗ trợ dịch chuyển đám mây. Kiến trúc *Microservices*, có bản chất của điện toán phân tán, kết hợp hoàn hảo với điện toán đám mây. Tổ

chức có thể triển khai đầy đủ các ứng dụng microservices vào đám mây công cộng (public cloud), riêng tư (private cloud) hoặc thậm chí là hỗn hợp (hybrid cloud). Có một thực tế là nhiều tổ chức đã triển khai các hệ thống cốt lõi dưới dạng kiến trúc microservices, chẳng hạn như Amazon, Best Buy, eBay, Netflix, Spotify và Uber, v.v. Vì các tổ chức có thể đã áp dụng tư duy microservices trong toàn bộ hệ thống CNTT của họ, việc triển khai ứng dụng BPC dựa trên kiến trúc microservices sẽ đơn giản hóa việc tích hợp BPC vào hệ thống tổ chức hiện có, do đó tiết kiệm thời gian và giảm chi phí triển khai. Việc xây dựng BPC framework theo kiến trúc microservices sẽ đảm bảo sự cập nhật và đón đầu xu hướng công nghệ đám mây.

### 3. Nghiên cứu có liên quan về quản trị tuân thủ quy trình nghiệp vụ

Phần này cung cấp một đánh giá tổng quan về các nghiên cứu liên quan đến lĩnh vực quản trị tuân thủ quy trình nghiệp vụ. Các nghiên cứu liên quan đến phương pháp BPC có thể được chia thành ba chủ đề chính: (1) Phương pháp BPC dựa trên môi trường triển khai tại chỗ (on-premises); (2) Phương pháp BPC dựa trên kiến trúc Service-Oriented-Architecture (SOA); (3) Phương pháp BPC dựa trên môi trường đám mây thông thường (general cloud).

Trong phương pháp tuân thủ quy trình nghiệp vụ dựa trên môi trường triển khai tại chỗ, temporal logic, deontic và defeasible logic, Petri nets, và workflow graphs là những ngôn ngữ mô hình hóa chính thức được sử dụng để phân tích sự tuân thủ của các quy trình nghiệp vụ [15], [21]–[25]. Goedertier và Vanthienen giới thiệu ngôn ngữ PENELOPE để xác minh và xác thực một tập hợp các nghĩa vụ thực thi được phân công (*deontic assignments*) [21]. Ngôn ngữ ConDec được đề xuất bởi Pesic và van der Aalst nhằm mô hình hóa và thực thi các quy trình nghiệp vụ [22]. Bên cạnh đó, Ghose và Koliadis giới thiệu một khuôn khổ đánh giá sử dụng cách tiếp cận theo kinh nghiệm (heuristic) để xác minh tình trạng tuân thủ của các quy trình nghiệp vụ [26]. Đề trực quan hóa các câu lệnh truy vấn, ngôn ngữ truy vấn (BPMN-Q) được xây dựng bởi Awad và cộng sự nhằm kiểm tra sự tuân thủ về mặt cấu trúc của các quy trình nghiệp vụ [23]. Ngoài ra, Governatori và Rotolo giới thiệu Process Compliance Language (PCL) nhằm cung cấp nền tảng khái niệm vững chắc cho suy luận và mô hình hóa cho các quy phạm pháp luật [24]. Nghiên cứu của Ly và cộng sự đưa ra SEAFLOWS như là một khung tuân thủ quy trình nghiệp vụ nhằm kiểm tra tuân thủ ngay trong giai đoạn thiết kế quy trình [25]. SEAFLOWS có thể kiểm tra tuân thủ với các luật định sẵn. Công cụ BRECA được xây dựng bởi Hà và cộng sự như là một kiến trúc kiểm tra sự tuân thủ nhằm khai phá các mối quan hệ hành vi của các quy trình nghiệp vụ và các yêu cầu quy phạm [15]. BRECA sử dụng cách tiếp cận mới PBR [9] để đạt được hiệu quả tính toán cao. Lưu ý rằng, các bộ công cụ để phục vụ hoạt động kiểm tra tuân thủ ở những nghiên cứu nêu trên đều được thiết kế cho môi trường on-premises.

Phương pháp tuân thủ quy trình nghiệp vụ dựa trên kiến trúc SOA tập trung vào việc đề xuất các công cụ hoặc frameworks được thiết kế với kiến trúc hướng dịch vụ (Service-oriented Architecture). Ví dụ điển hình là các kỹ

thuật kiểm tra tuân thủ dựa trên các ngôn ngữ mô hình hóa như: temporal/ deontic logic [27], [28], business vocabulary, và business rules/ rule interchange format [29], first-order-logic/ finite state machines [30]. Bộ công cụ COMPAS được Elgammal và cộng sự giới thiệu nhằm cung cấp khả năng quản trị tuân thủ quy trình nghiệp vụ một cách toàn diện [27]. COMPAS có thể xác minh tính tuân thủ trong cả vòng đời của quy trình nghiệp vụ. Nghiên cứu này tiến hành phân loại các ràng buộc tuân thủ dựa trên các mẫu cho các quy trình nghiệp vụ. Sau đó, các phân tích được tiến hành để giải quyết các vi phạm tuân thủ ngay trong quá trình thiết kế quy trình nghiệp vụ [28]. Bên cạnh đó, các đặc điểm hoàn chỉnh của các tập luật phục vụ cho quá trình kiểm tra tuân thủ chính sách kinh doanh dựa trên kiến trúc SOA được cung cấp bởi Weigand và cộng sự [29]. Đề xuất này đặc biệt hữu ích cho việc kiểm tra sự tuân thủ về mặt cấu trúc của các quy trình nghiệp vụ. Tương tự như Weigand và cộng sự [29], Rodríguez và cộng sự [30] đề xuất một khuôn khổ quản trị tuân thủ dựa trên kiến trúc SOA để đánh giá sự tuân thủ của các quy trình.

Phương pháp tuân thủ quy trình nghiệp vụ dựa trên kiến trúc đám mây thông thường tập trung vào các kỹ thuật kiểm tra tuân thủ được xây dựng dựa trên kiến trúc khách-chủ (client-server) truyền thống. Ngoại trừ nghiên cứu của Accorsi và cộng sự là sử dụng ngôn ngữ Petri nets [32], những nghiên cứu còn lại trong nhóm này không đề cập đến ngôn ngữ mô hình hóa quy trình được sử dụng để kiểm tra tuân thủ quy trình nghiệp vụ [33]–[35]. Ở nhóm chủ đề này, một loạt các công cụ phục vụ cho hoạt động BPC đã được đề xuất. Comcert, được giới thiệu bởi Accorsi và cộng sự, là một công cụ để phân loại các quy tắc tuân thủ từ các khuôn khổ quy định cho quy trình nghiệp vụ dựa trên nền tảng đám mây [32]. Các quy tắc (rules) này sau đó được chuẩn hóa trong Petri nets nhằm tự động phát hiện các hành vi không tuân thủ. Compagna và cộng sự đề xuất SVaaS như là một công cụ để xác định sự tuân thủ của các quy trình nghiệp vụ ở giai đoạn thiết kế [33]. SVaaS cung cấp giao diện đồ họa trực quan phục vụ việc kiểm tra tuân thủ quy trình nghiệp vụ ở môi trường máy chủ. Tương tự như Accorsi và cộng sự, Elgammal và Turetken cũng đề xuất một khuôn khổ để quản trị sự tuân thủ quy trình nghiệp vụ nhưng bằng cách sử dụng lý thuyết bản thể học (ontology) với trọng tâm chính là kho lưu trữ tuân thủ [34]. Nó cho phép sao lưu toàn bộ vòng đời của quá trình kiểm tra tuân thủ quy trình nghiệp vụ. Ngoài ra, Singh và Sidhu đề xuất hệ thống CMTES nhằm đánh giá và xác định độ tin cậy của các nhà cung cấp dịch vụ đám mây (CSP) dựa trên hoạt động kiểm tra tuân thủ trên các khía cạnh khác nhau [35].

Từ các nghiên cứu liên quan ở trên, chúng ta có thể thấy, các BPC frameworks dựa môi trường triển khai truyền thống đã được đề xuất nhiều hơn so với môi trường đám mây. Nhận định trên được khẳng định bởi nghiên cứu gần đây của Mustapha và cộng sự [7]. Đối với nhóm kiến trúc SOA, mặc dù nó được sử dụng khá nhiều trong hoạt động kiểm tra tuân thủ quy trình nghiệp vụ, tuy nhiên việc sử dụng SOA lại tồn tại một số hạn chế nhất định (xem phần 2). Theo hiểu biết của nhóm tác giả, hiện chưa có phương pháp BPC nào được thiết kế cho kiến trúc microservice. Tuy nhiên, một thực tế được chỉ ra bởi nghiên cứu của Nordemann và cộng sự là kiến trúc



microservices đã và đang được ứng dụng trong nhiều khía cạnh của BPM [36]. Do BPC được cấu thành từ BPM, nên không thể nằm ngoài xu thế. Kết quả của quá trình nghiên cứu tài liệu liên quan đã chỉ ra một khoảng trống nghiên cứu trong việc áp dụng kiến trúc microservices vào BPC.

#### 4. BPC4MSA: Kiến trúc microservices dựa trên nền tảng đám mây trong quản trị tuân thủ quy trình nghiệp vụ

Một trong những thách thức của quá trình phát triển và triển khai kiến trúc microservice là việc phân rã hệ thống thành các dịch vụ nhỏ hơn. Có nhiều chiến lược và cách tiếp cận khác nhau cho quá trình này. Đối với nghiên cứu này, bài báo sẽ áp dụng cách tiếp cận phân rã hệ thống dựa trên *chức năng nghiệp vụ* (business capabilities) và theo *trường hợp sử dụng* (use cases) [20]. Cách tiếp cận này phân tách hệ thống theo chiều dọc, từ đó tạo ra những hệ thống con (subsystems) có khả năng hoạt động độc lập ở một mức độ nhất định.

##### 4.1. Tổng quan về kiến trúc BPC được đề xuất

Trong phần này, bài báo đề xuất kiến trúc tích hợp giữa BPC framework (BRECA) và microservice (MSA). Vì kiến trúc đề xuất tập trung vào BPC, nhóm tác giả đặt tên cho kiến trúc này là BPC4MSA. Hệ thống BPC truyền thống (Hình 1) sẽ được phân rã theo các chức năng và tài nguyên dựa trên chiến lược phân rã đã đề cập. Một dịch vụ (service) sẽ bao gồm tất cả các hoạt động độc lập liên quan đến một tài nguyên cụ thể (Ví dụ: Mô hình Quy trình nghiệp vụ từ không gian PAIS, các quy tắc tuân thủ được mã hóa từ không gian tuân thủ). Hình 3 mô tả kiến trúc của BPC4MSA dựa trên khung BPC (BRECA) được đề xuất bởi Hà và cộng sự [9], [15]. Dưới đây là mô tả tổng quan về hệ thống BPC4MSA:

Các dịch vụ độc lập được nhóm thành từng cụm phòng theo từng loại hình và chức năng cụ thể.

- *Cụm dịch vụ tự động hóa BPC*: Cụm này sẽ đảm nhận công việc kiểm tra tính tuân thủ của các quy trình nghiệp vụ.

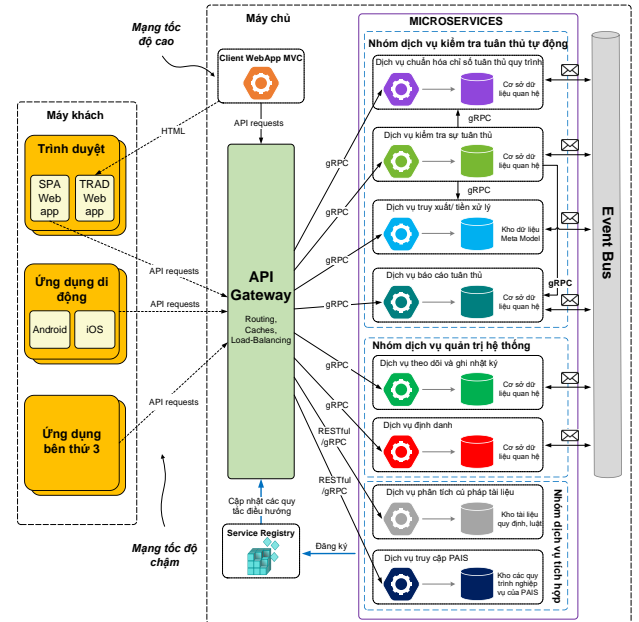
- *Cụm dịch vụ quản trị hệ thống*: Cụm này có hai dịch vụ quản lý. Cái đầu tiên phục vụ mục đích giám sát và ghi nhật ký (logging) hệ thống nhằm theo dõi báo cáo lỗi và lưu trữ dữ liệu liên quan một cách tập trung. Dịch vụ thứ hai được sử dụng để đảm bảo rằng chỉ những người dùng đã được xác thực mới có thể truy cập vào các chức năng cụ thể mà họ được ủy quyền.

- *Cụm dịch vụ tích hợp bên ngoài*: Các dịch vụ tích hợp sẽ đóng vai trò trung gian để giao tiếp với các hệ thống bên ngoài như: (1) PAIS để truy xuất các mô hình quy trình và các siêu dữ liệu (metadata) liên quan và (2) các công cụ phân tích cú pháp (parsing tools) phục vụ quá trình chuyển đổi các văn bản hoặc tài liệu pháp lý hoặc quy định (regulations) thành các nội dung máy tính có thể hiểu được (machine-readable).

Toàn bộ kiến trúc được chia thành hai phần, máy khách (client-side) và máy chủ (server-side). Mặc dù, có thể tương tác trực tiếp từng dịch vụ của máy chủ từ máy khách, kiến trúc được đề xuất trong bài báo này lựa chọn sử dụng một API gateway để tập trung hóa việc giao tiếp giữa máy khách và các dịch vụ. Tất cả thành phần của BPC4MSA đều được giao tiếp thông qua các API, hỗ trợ

cả giao tiếp gRPC (Google Remote Procedure Call) hiệu suất cao và RESTful API tiêu chuẩn.

Như thể hiện trong Hình 3, BPC4MSA hỗ trợ một loạt các chức năng của BPC truyền thống (xem Hình 1). Ngoài ra, để đáp ứng các yêu cầu của kiến trúc microservice, mỗi dịch vụ của BPC4MSA chạy phi tập trung, độc lập và có cơ sở dữ liệu riêng. Thiết kế này dẫn đến sự *khớp nối lỏng lẻo* (loose coupling) giữa các dịch vụ từ đó cho phép sử dụng một công nghệ phù hợp cho từng dịch vụ cụ thể. Về mặt triển phần mềm, mỗi nhóm phát triển dịch vụ có thể chọn những nhóm công nghệ (ví dụ: C#, Java, Python, JavaScript, PHP, v.v.) phù hợp với nhu cầu của mình.



Hình 3. Tổng quan kiến trúc của BPC4MSA

Đề đơn giản hóa trong việc trình bày, không phải tất cả các dịch vụ có sẵn đều được hiển thị trong Hình 3. Tuy nhiên, các dịch vụ vẫn có thể được bổ sung nhằm mở rộng BPC4MSA một cách phù hợp. Đó có thể được coi là lợi thế của mô hình thiết kế microservices, nơi chúng ta có thể thêm nhiều chức năng hơn vào hệ thống một cách độc lập và nhanh chóng mà ít ảnh hưởng tới kiến trúc tổng thể của hệ thống. Cần lưu ý rằng, tất cả các dịch vụ của BPC4MSA không yêu cầu phải được lưu trữ trong cùng một máy chủ vật lý hoặc cùng một nhà cung cấp dịch vụ hay hạ tầng đám mây duy nhất.

BPC4MSA cho phép người dùng phân tích và mô phỏng các đặc điểm của quy trình nghiệp vụ dựa trên các thuộc tính khác nhau. Kết quả phân tích có thể được tổng hợp bằng các dịch vụ báo cáo (reporting services) cung cấp nền tảng cho chức năng của trang tổng quan (dashboard).

##### 4.2. API gateway

API gateway đóng một vai trò thiết yếu trong việc điều phối và quản lý luồng thông tin giữa các dịch vụ và các bên liên quan. API gateway, một phần mềm trung gian nằm giữa một điểm cuối API và các dịch vụ phụ trợ, cung cấp các công nghệ giao tiếp nhẹ (lightweight communication technologies), cho phép các microservices giao tiếp với nhau khác trên cơ sở hạ tầng vật lý một cách nhanh chóng. API Gateway cũng được kết hợp với các mô-đun *cân bằng tải* (load balancing), *bộ nhớ đệm* (caching) và *định tuyến*

(routing) để quản lý lưu lượng truy cập của người dùng (xem Hình 3).

Như mô tả ở Hình 3, giao tiếp giữa máy khách và máy chủ được triển khai bằng API RESTful thông qua API gateway. Từ đó giúp giảm số lượng cuộc gọi (đồng thời) dịch vụ và do đó tăng hiệu suất hệ thống [20]. Tuy nhiên, trái ngược với các ESB phức tạp của SOA, công API không cung cấp bất kỳ logic định tuyến phức tạp.

#### 4.3. Tìm kiếm microservices

Trong môi trường lưu trữ động (dynamic hosting) của microservices, mỗi dịch vụ có thể có một địa chỉ IP khác nhau. Do đó, ta không thể cấu hình tĩnh một máy khách với địa chỉ IP của các dịch vụ. Thay vào đó, hệ thống phải sử dụng cơ chế *tìm kiếm các dịch vụ* (services discovery) một cách tự động. Cơ chế hoạt động của services discovery khá đơn giản: Thành phần chính của nó là *sổ đăng ký dịch vụ* (service registry), đây là một cơ sở dữ liệu về các vị trí mạng của các dịch vụ [8]. Cơ chế services discovery cập nhật service registry khi các dịch vụ bắt đầu hoạt động hoặc dừng. Khi một máy khách gọi một dịch vụ, cơ chế tìm kiếm dịch vụ sẽ truy vấn sổ đăng ký dịch vụ để lấy danh sách các phiên bản dịch vụ có sẵn và định tuyến yêu cầu đến một trong số các dịch vụ này. Service Registry sẽ hoạt động như một hệ thống đăng ký cho microservices. Service Registry có thể được tham vấn bởi API Gateways và microservices để tìm kiếm dịch vụ. Sổ đăng ký này sẽ đóng vai trò là kho lưu trữ chứa siêu dữ liệu (metadata) của tất cả các microservices.

#### 4.4. Ứng dụng máy khách

Các ứng dụng phía máy khách (client-side applications) chủ yếu giao tiếp với các microservices từ phía máy chủ thông qua API gateway. Mẫu thiết kế máy khách - chủ thực thi sự tách biệt các mối quan tâm, giúp máy khách và các thành phần máy chủ phát triển độc lập với nhau. Giao tiếp giữa các ứng dụng khách và microservices từ máy chủ được thiết lập thông qua RESTful API. REpresentational State Transfer (REST) là một phong cách kiến trúc hướng tài nguyên được trình bày vào năm 2000 trong luận án nổi tiếng của Roy Fielding [37]. Trong khi đó, RESTful API đề cập đến *giao diện lập trình ứng dụng* (API) tuân theo kiểu kiến trúc REST.

Các ứng dụng phía máy khách yêu cầu tài nguyên từ máy chủ chủ yếu thông qua các RESTful API. Điểm cuối (endpoint) của các API này, là một đầu của kênh giao tiếp (URL), được tổng hợp tại API gateway (như trong Hình 3). Vì máy chủ cung cấp các chức năng của nó thông qua các API, các loại ứng dụng máy khách cho BPC4MSA bao gồm (nhưng không giới hạn):

- *Ứng dụng web*: Có hai cách tiếp cận chung để xây dựng ứng dụng web, ứng dụng web truyền thống (TRAD) và ứng dụng web Single-Page Application (SPA) [38]. Trong khi ứng dụng web TRAD thực hiện hầu hết logic ứng dụng trên máy chủ, SPA thực hiện hầu hết logic giao diện người dùng trong trình duyệt web, giao tiếp với máy chủ chủ yếu bằng các RESTful API. Các ứng dụng web truyền thống (TRAD) xử lý các yêu cầu HTTP từ các trình duyệt, lấy các trang HTML từ dịch vụ Client WebApp MVC và thường cung cấp các ứng dụng máy khách đơn giản và chỉ đọc. Đối với SPA, nhiều công nghệ hiện đại có thể triển khai, chẳng hạn như JavaScript, TypeScript hoặc

Blazor WebAssembly [38] nhằm tăng tốc độ phản hồi và tăng cường trải nghiệm người dùng.

- *Ứng dụng di động*: Loại này sử dụng các API được xây dựng riêng cho các ứng dụng di động. Nhu cầu về ứng dụng di động khác với ứng dụng web. Đối với ứng dụng di động, các API có thể cần phải tối ưu hóa hơn nữa để cung cấp sự phản hồi dữ liệu hiệu quả hơn. Ứng dụng dành cho thiết bị di động cần giao tiếp với công API để yêu cầu *dữ liệu tổng hợp* (aggregate data) từ nhiều microservices phía sau API gateway và trả về một tập dữ liệu duy nhất, loại bỏ bất kỳ dữ liệu không cần thiết trong phản hồi. Dữ liệu trả về có thể được nén để giảm tiêu thụ băng thông. Do đó, nó làm giảm lưu lượng mạng và cải thiện thời gian phản hồi của máy chủ.

- *Ứng dụng của bên thứ ba*: Máy chủ cũng cung cấp các API công khai cho các nhà phát triển bên thứ ba (third-party). Các ứng dụng của bên thứ ba có thể truy cập các API công khai qua internet, do đó, thành phần API có thể không tối ưu về mặt thông tin trả về [8]. Nhưng sự kém hiệu quả của thành phần API là một vấn đề tương đối nhỏ so với thách thức quan trọng hơn nhiều trong việc thiết kế một API phù hợp mà mỗi ứng dụng của bên thứ ba sử dụng. Đó là bởi vì các nhà phát triển bên thứ ba cần một API ổn định để tích hợp với hệ thống của họ. Nói chung, các API cho ứng dụng của bên thứ ba thường có mục đích chung chung không được tối ưu hóa cho từng trường hợp sử dụng, như ứng dụng web hoặc di động.

#### 4.5. Service cho việc tự động hóa kiểm tra tuân thủ

Các dịch vụ phục vụ cho các tác vụ tự động hóa kiểm tra tuân thủ quy trình nghiệp vụ là thành phần cốt lõi của BPC4MSA. Thành phần này bao gồm bốn dịch vụ chính. Chúng đảm bảo rằng, chức năng kiểm tra tuân thủ hoạt động một cách chính xác. Giao tiếp giữa các dịch vụ này được thiết lập thông qua giao thức gRPC hiệu suất cao (xem Hình 3). Mỗi dịch vụ cũng báo cáo trạng thái hoặc sự kiện liên quan với nó cho bus sự kiện (event bus) để các dịch vụ khác có thể nhận được thông báo khi cần thiết. Chức năng chi tiết của bốn dịch vụ trong cụm này bao gồm:

- *Dịch vụ chuẩn hóa chỉ số tuân thủ quy trình* (compliance metric formalization): Dịch vụ này truy xuất các *điều kiện cần phải tuân thủ* ở định dạng máy tính có thể đọc được từ dịch vụ *phân tích cú pháp* của các văn bản quy phạm thông qua API gateway. Sau đó, các quy tắc này được chuyển đổi thành các chỉ số tuân thủ (compliance metrics), ví dụ như *mối quan hệ hành vi* (Behavioral Relations – BHR [9]) và sẵn sàng cung cấp cho dịch vụ kiểm tra tuân thủ quy trình nghiệp vụ.

- *Dịch vụ truy xuất/ tiền xử lý*: Dịch vụ này truy xuất các mô hình quy trình nghiệp vụ và các meta data liên quan từ dịch vụ truy cập PAIS thông qua API gateway. Sau đó, các mô hình nghiệp vụ và thông tin liên quan sẽ được xử lý trước (nếu cần) và sẵn sàng sử dụng.

- *Dịch vụ kiểm tra sự tuân thủ*: Dịch vụ này được hỗ trợ bởi phương pháp tính toán tiên tiến, có tên là *Partitioning Behavioral Retrieval* (PBR), được đề xuất bởi Hà và Prinz [9]. Tất cả sức mạnh xử lý của BPC4MSA sẽ tập trung vào dịch vụ này. Dịch vụ kiểm tra sự tuân thủ nhận BHR làm tham số đầu vào từ *dịch vụ chuẩn hóa chỉ số tuân thủ quy trình* và thông tin về các mô hình quy trình nghiệp vụ từ

*dịch vụ truy xuất/ tiền xử lý*. Sau đó, nó đưa ra những suy luận về mức độ tuân thủ của mô hình quy trình đã cho dựa trên các quy tắc tuân thủ bắt buộc được mã hóa trong BHR. Giao tiếp giữa các services trên cũng được thực hiện bằng gRPC thông qua giao tiếp trực tiếp giữa các dịch vụ hoặc thông qua API gateway. Sau đó, đầu ra của dịch vụ này có thể được chuyển tiếp tới *dịch vụ báo cáo Tuân thủ*.

- *Dịch vụ báo cáo tuân thủ*: Dịch vụ này tạo và kết xuất các báo cáo dựa trên nhu cầu của máy khách. Đầu vào của dịch vụ này được truy xuất từ *dịch vụ kiểm tra sự tuân thủ* thông qua giao thức gRPC.

#### 4.6. Services quản trị hệ thống

Hai dịch vụ chính cho phép quản lý toàn bộ hệ thống BPC4MSA. Thứ nhất, *dịch vụ định danh* (identity service) cho phép máy chủ triển khai các biện pháp bảo vệ cần thiết và cung cấp quyền truy cập vào các microservice cho máy khách được ủy quyền. Dịch vụ này có thể bao gồm việc triển khai *mã hóa tin nhắn* (message encryptions), *mã thông báo truy cập* (access tokens) hoặc các tiến trình xử lý (transaction) để ủy quyền và xác thực. Dịch vụ này giúp xác thực máy khách và hệ thống, xác thực giữa các microservices và xác thực của bên thứ ba.

Thứ hai, *dịch vụ theo dõi và ghi nhật ký* (monitoring and logging) cho phép BPC4MSA kiểm tra tình trạng hoạt động của các microservices và toàn bộ hệ thống. Dịch vụ này bao gồm việc kiểm tra hoạt động trên các microservice và tình hình hoạt động của cả hệ thống. Ngoài ra, nó cũng cung cấp chức năng phân tích, hiển thị số liệu thống kê như thời gian chạy và hoạt động (up-time) của microservices.

#### 4.7. Event Bus

*Bus sự kiện* (event bus) cho phép các microservices giao tiếp kiểu publish/subscribe mà không yêu cầu các dịch vụ phải biết về nhau. Chúng ta cũng có thể mô tả định dạng của các thông điệp được trao đổi bằng cách sử dụng một số định dạng dữ liệu tiêu chuẩn như JSON, XML hoặc Protobuf. Mô hình *sự kiện* (event) và *chủ đề* (topics) của thông điệp là một sự trừu tượng và là một cách hoàn hảo để thiết kế API bất đồng bộ (asynchronous) cho một dịch vụ. Nhưng để triển khai tốt một dịch vụ, chúng ta cần chọn một công nghệ *event bus* và xác định cách triển khai theo thiết kế. Một số ứng dụng dựa trên tin nhắn thường sử dụng một message broker, một dịch vụ cơ sở hạ tầng mà qua đó các dịch vụ này giao tiếp với nhau [8]. Message broker là nơi trung gian mà thông qua đó tất cả các tin nhắn đều được lưu chuyển.



Hình 4. Sơ đồ hoạt động của Bus sự kiện

Như được minh họa trong Hình 4, microservice A xuất bản (publish) lên Bus sự kiện, phân phối tới các microservices B và C đã đăng ký. Lưu ý rằng, microservice A không nhận biết được những bên thụ hưởng thông tin.

Có nhiều cách triển khai cho event bus, ta có thể sử

dụng công nghệ/ hạ tầng của RabbitMQ, Azure Service Bus, Google Cloud Pub/ Sub hoặc bất kỳ phần mềm nguồn mở hoặc thương mại nào của bên thứ ba.

## 5. Kết luận

Việc tuân thủ quy định đã được nghiên cứu trong các lĩnh vực khác, chẳng hạn như luật và lý luận pháp lý (legal reasoning). Tuy nhiên, người ta thường ít chú ý hơn đến quản trị quy trình nghiệp vụ và kiến trúc microservices dựa trên nền tảng đám mây. Trong xu thế chuyển đổi số, dịch vụ chuyển qua đám mây đã trở thành một xu hướng không thể đảo ngược. Trên thực tế, kiến trúc microservices đã và đang được các doanh nghiệp và tổ chức áp dụng với tốc độ ngày càng cao.

Các tài liệu liên quan về các khuôn khổ và công cụ tuân thủ quy trình nghiệp vụ đã được tiến hành đánh giá một cách có hệ thống. Trong đó, hầu hết các phương pháp tiếp cận trước đây đều tập trung vào việc đề xuất khuôn khổ tuân thủ cho các môi trường hoạt động truyền thống (on-premise), trong khi các cách tiếp cận khác tập trung vào SOA dựa trên đám mây. Do đó, nó chỉ ra nhu cầu về một BPC framework cho kiến trúc microservice trong môi trường hoạt động dựa trên nền tảng điện toán đám mây.

BPC4MSA là framework đầu tiên cung cấp một khung kiểm tra tuân thủ quy trình nghiệp vụ với đầy đủ chức năng và được thiết kế dựa trên kiến trúc microservices. BPC4MSA được thiết kế dựa trên khung BPC truyền thống (BRECA), một trong những khung hỗ trợ hoạt động kiểm tra tuân thủ một cách hiệu quả và nhanh chóng [15]. Bất chấp những ưu điểm của việc sử dụng kiến trúc microservice, một số vấn đề vẫn cần được xem xét. Ví dụ, (1) cách thức phân rã một cách hiệu quả một hệ thống monolithic thành các microservices và (2) cách thức quản trị dữ liệu thống nhất trên các cơ sở dữ liệu phân tán, cả hai đều không phải là việc dễ dàng. Tuy nhiên, việc sử dụng microservices vẫn mang lại nhiều ưu điểm hơn là nhược điểm.

Đối với các cơ hội nghiên cứu trong tương lai, nhóm tác giả có một số đề xuất như sau. Đầu tiên, nhóm tác giả dự định xây dựng một hệ thống BPC4MSA (dạng proof of concept) nhằm đánh giá mức độ khả thi trong môi trường thực tế. Hơn nữa, một nghiên cứu thực nghiệm (empirical study) có thể được triển khai nhằm so sánh các hệ thống BPC khác nhau, từ đó cung cấp những gợi ý cho các nhà nghiên cứu và thực hành trong bối cảnh chuyển đổi số và dịch vụ chuyển lên đám mây.

## TÀI LIỆU THAM KHẢO

- [1] L. D. Binh and T. T. Phuong, "Digital Economy And Digital Transformation in Vietnam", Hanoi, 2020.
- [2] C. Möller, C. J. Maack, and R. D. Tan, "What is Business Process Management: A Two Stage Literature Review of an Emerging Field", in *Research and Practical Issues of Enterprise Information Systems II Volume 1*, Boston, MA: Springer US, 2007, pp. 19–31. doi: 10.1007/978-0-387-75902-9\_3.
- [3] J. B. Hill, J. Sinur, D. Flint, and M. J. Melenovsky, "Gartner's position on business process management", 2006. [Online]. Available: <https://www.gartner.com/en/documents/489533>.
- [4] M. el Kharbili, A. K. A. de Medeiros, S. Stein, and W. M. P. van der Aalst, "Business process compliance checking: Current state and future challenges", in *Modellierung betrieblicher Informationssysteme (MobIS 2008)*, 2008, pp. 107–113.

- [5] G. Governatori and S. Sadiq, "The Journey to Business Process Compliance", *Handb. Res. Bus. Process Model.*, pp. 426–454, 2008, doi: 10.4018/978-1-60566-288-6.ch020.
- [6] Cameron *et al.*, "Vietnam's future digital economy - Towards 2030 and 2045", CSIRO, Brisbane, 2019.
- [7] A. M. Mustapha, O. T. Arogundade, S. Misra, R. Damasevicius, and R. Maskeliunas, "A systematic literature review on compliance requirements management of business processes", *Int. J. Syst. Assur. Eng. Manag.*, vol. 11, no. 3, pp. 561–576, 2020, doi: 10.1007/s13198-020-00985-w.
- [8] C. Richardson, *Microservices patterns: with examples in Java*. Simon and Schuster, 2018. [Online]. Available: <https://www.manning.com/books/microservices-patterns>
- [9] N. L. Ha and T. M. Prinz, "Partitioning Behavioral Retrieval: An Efficient Computational Approach With Transitive Rules", *IEEE Access*, vol. 9, pp. 112043–112056, 2021, doi: 10.1109/ACCESS.2021.3102634.
- [10] M. Dumas, M. La Rosa, J. Mendling, and H. A. Reijers, *Fundamentals of Business Process Management*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2018. doi: 10.1007/978-3-662-56509-4.
- [11] W. M. P. van der Aalst, A. Hirschsall, and H. M. W. Verbeek, "An Alternative Way to Analyze Workflow Graphs", in *CAiSE '02*, 2002, pp. 535–552. doi: 10.1007/3-540-47961-9\_37.
- [12] D. W. Amer, J. Barberis, R. P. Buckley, R. D. Arnott, and T. R. Aronson, *FinTech and RegTech in a Nutshell, and the Future in a Sandbox*. CFA Institute Research Foundation, 2017. doi: 10.2470/rfbr.v3.n4.1.
- [13] G. Governatori, "Representing Business Contracts In RuleML", *Int. J. Coop. Inf. Syst.*, vol. 14, no. 02n03, pp. 181–216, Jun. 2005, doi: 10.1142/S0218843005001092.
- [14] H. A. López, S. Debois, T. Slaats, and T. T. Hildebrandt, "Business process compliance using reference models of law", *Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics)*, vol. 12076 LNCS, pp. 378–399, 2020, doi: 10.1007/978-3-030-45234-6\_19.
- [15] N. L. Ha, T. T. Nguyen, and S. H. Do, "Ứng Dụng Công Nghệ Tuân Thủ Quy Trình Nghiệp Vụ Trong Hoạt Động Chuyển Đổi Số Của Ngành Tài Chính: Góc Nhìn Từ Fintech Và Regtech", *Tạp Chí Khoa học Quản lý và Kinh tế, Trường Đại học Kinh tế, Đại học Huế*, vol. 18, 2021, [Online]. Available: <https://tapchi.hce.edu.vn/index.php/sjme/article/view/73>
- [16] M. Hashmi, G. Governatori, H. P. Lam, and M. T. Wynn, "Are we done with business process compliance: state of the art and challenges ahead", *Knowl. Inf. Syst.*, vol. 57, no. 1, pp. 79–133, 2018, doi: 10.1007/s10115-017-1142-1.
- [17] Z. Fang and C. Yin, "BPM Architecture Design Based on Cloud Computing", *Intell. Inf. Manag.*, vol. 02, no. 05, pp. 329–333, 2010, doi: 10.4236/iim.2010.25039.
- [18] N. Dragoni *et al.*, "Microservices: Yesterday, Today, and Tomorrow", in *Present and Ulterior Software Engineering*, Cham: Springer International Publishing, 2017, pp. 195–216. doi: 10.1007/978-3-319-67425-4\_12.
- [19] N. Kratzke and P. C. Quint, "Understanding cloud-native applications after 10 years of cloud computing - A systematic mapping study", *J. Syst. Softw.*, vol. 126, pp. 1–16, 2017, doi: 10.1016/j.jss.2017.01.001.
- [20] S. Alpers, C. Becker, A. Oberweis, and T. Schuster, "Microservice Based Tool Support for Business Process Modelling", in *2015 IEEE 19th International Enterprise Distributed Object Computing Workshop*, Sep. 2015, pp. 71–78. doi: 10.1109/EDOCW.2015.32.
- [21] S. Goedertier and J. Vanthienen, "Designing Compliant Business Processes with Obligations and Permissions", in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 4103 LNCS, 2006, pp. 5–14. doi: 10.1007/11837862\_2.
- [22] M. Pesic and W. M. P. van der Aalst, "A Declarative Approach for Flexible Business Processes Management", in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 4103 LNCS, no. September 2006, 2006, pp. 169–180. doi: 10.1007/11837862\_18.
- [23] A. Awad, M. Weidlich, and M. Weske, "Visually specifying compliance rules and explaining their violations for business processes", *J. Vis. Lang. Comput.*, vol. 22, no. 1, pp. 30–55, 2011, doi: 10.1016/j.jvlc.2010.11.002.
- [24] G. Governatori and A. Rotolo, "A conceptually rich model of business process compliance", *Conf. Res. Pract. Inf. Technol. Ser.*, vol. 110, pp. 3–12, 2010.
- [25] L. T. Ly *et al.*, "SeaFlows Toolset – Compliance Verification Made Easy for Process-Aware Information Systems", in *Lecture Notes in Business Information Processing*, vol. 72 LNBIP, no. May 2014, 2011, pp. 76–91. doi: 10.1007/978-3-642-17722-4\_6.
- [26] A. Ghose and G. Koliadis, "Auditing Business Process Compliance", in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 4749 LNCS, 2007, pp. 169–180. doi: 10.1007/978-3-540-74974-5\_14.
- [27] A. Elgammal, O. Turetken, W.-J. van den Heuvel, and M. Papazoglou, "On the Formal Specification of Regulatory Compliance: A Comparative Analysis", in *Service-Oriented Computing*, E. M. Maximilien, G. Rossi, S.-T. Yuan, H. Ludwig, and M. Fantinato, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2011, pp. 27–38. doi: 10.1007/978-3-642-19394-1\_4.
- [28] A. Elgammal, O. Turetken, and W.-J. Van Den Heuvel, "Using Patterns for the Analysis and Resolution of Compliance Violations", *Int. J. Coop. Inf. Syst.*, vol. 21, no. 01, pp. 31–54, Mar. 2012, doi: 10.1142/S0218843012400023.
- [29] H. Weigand, W.-J. van den Heuvel, and M. Hiel, "Business policy compliance in service-oriented systems", *Inf. Syst.*, vol. 36, no. 4, pp. 791–807, Jun. 2011, doi: 10.1016/j.is.2010.12.005.
- [30] C. Rodríguez, D. Schleicher, F. Daniel, F. Casati, F. Leymann, and S. Wagner, "SOA-enabled compliance management: Instrumenting, assessing, and analyzing service-based business processes", *Serv. Oriented Comput. Appl.*, vol. 7, no. 4, pp. 275–292, 2013, doi: 10.1007/s11761-013-0129-3.
- [31] T. E. Abioye, O. T. Arogundade, S. Misra, K. Adesemowo, and R. Damaševičius, "Cloud-based business process security risk management: A systematic review, taxonomy, and future directions", *Computers*, vol. 10, no. 12, pp. 1–28, 2021, doi: 10.3390/computers10120160.
- [32] R. Accorsi, L. Lewis, and Y. Sato, "Automated Certification for Compliant Cloud-based Business Processes", *Bus. Inf. Syst. Eng.*, vol. 3, no. 3, pp. 145–154, Jun. 2011, doi: 10.1007/s12599-011-0155-7.
- [33] L. Compagna, P. Guilleminot, and A. D. Brucker, "Business process compliance via security validation as a service", *Proc. - IEEE 6th Int. Conf. Softw. Testing, Verif. Validation, ICST 2013*, pp. 455–462, 2013, doi: 10.1109/ICST.2013.63.
- [34] A. Elgammal and O. Turetken, "Lifecycle Business Process Compliance Management: A Semantically-Enabled Framework", in *2015 International Conference on Cloud Computing (ICCC)*, Apr. 2015, no. March, pp. 1–8. doi: 10.1109/CLOUDCOMP.2015.7149646.
- [35] S. Singh and J. Sidhu, "Compliance-based Multi-dimensional Trust Evaluation System for determining trustworthiness of Cloud Service Providers", *Futur. Gener. Comput. Syst.*, vol. 67, pp. 109–132, Feb. 2017, doi: 10.1016/j.future.2016.07.013.
- [36] F. Nordemann, R. Tönjes, E. Pulvermüller, and H. Tapken, "Resilient Business Process Modeling and Execution Using BPMN and Microservices", in *Communications in Computer and Information Science*, vol. 1361, 2021, pp. 175–199. doi: 10.1007/978-3-030-67445-8\_8.
- [37] R. T. Fielding, *Architectural styles and the design of network-based software architectures*. University of California, Irvine, 2000.
- [38] Microsoft, "Choose Between Traditional Web Apps and Single Page Apps (SPAs)", *Microsoft Docs*, 2021. <https://docs.microsoft.com/en-us/dotnet/architecture/modern-web-apps-azure/choose-between-traditional-web-and-single-page-apps> (accessed Mar. 24, 2022).