IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

ISO 3297:2007 Certified ⁑ Impact Factor 8.102 ⁑ Peer-reviewed / Refereed journal ⁑ Vol. 12, Issue 6, June 2023

DOI: 10.17148/IJARCCE.2023.12668

Cloud ERP Customization using Serverless Runtime

Naveen Trivedi[1] and Vijay Mohan Shrimal[2]

M.Tech Research Scholar, Jagannath University Jaipur, Rajasthan, India[1]

Assistant Professor, Jagannath University Jaipur, Rajasthan, India[2]

Abstract: With the advent of cloud computing, it has been proved to be quite useful technology for development and delivery of ERP software. There is often need to customize an ERP software to fit the business process variation of an organization. Many research studies published during 2013 to 2022 acknowledges that deep customization of Cloud ERP System continue to be major challenge especially with public cloud-based edition. There is lack of further study to use server-less architecture to address deep customization needs. This study will present use of open FaaS to call function's client specific implementation via API gateway so to provide deep customization capability in ERPsoftware. This is supported by a proof of concept with reference application of pricing element determination.

Keyword: ERP, Customization, SaaS, Server-less runtime, Function-as-a-Service

I. INTRODUCTION

ERP refers to enterprise resource planning. An ERP system consists of various software modules to handle different business process of an organization. Some of basic modules of an ERP system are CRM, Master Data Management, HCM, SCM, Material Management, Sales & Distribution, Finance & Controlling, Product Life cycle Management, Production Planning, Project Systems, Plant Maintenance etc. Each of these modules caters to different business process within the scope the given module. E.g. SD handles OTC (Order to Cash) & P2P (Procure to Pay) related business process. Various business processes in these modules differ from company to company and industry to industry.

Thus, an ERP system had to address wide ranges of complexity because various factor, some of those in interest of this study are listed below:

- ➢ Complexity of business process
- ➢ Difference in business process of different organization of same industry
- ➢ Different industry verticals.

Traditional ERP system uses different techniques to handle above by providing either different industry specific ERP solution and/or different customization approaches so that customer using the ERP system can customize the software during implementation phase. These customization needs vary depending upon each process and how generically it's implemented. Typically, Implementation of ERP software require somewhere between few months to couple of years without guarantee of success.

Thus, a typical ERP software delivery requires a product development by ERP vendor in such a way that it can cover the basic business process of a focused industry or industries. Following which once an organization choose to use the ERP software, they do an implementation of ERP software which mainly focuses on mapping the requirement to given ERP software, exact match business process is used as it from the ERP software, gaps found are then fulfilled by either means of customizing the product or developing custom functions in ERP system or by some external means. Once the implementation phase is over, the software is then rolled to business segments depending on roll out strategy.

In this study "ERP vendor" term is used to refer an IT organization which develops ERP software, Consulting partner term is used to refer to an "IT organization" which assist in implementation of ERP software and term "Customer" or "Organization" refers to an organization which uses an ERP software.

Providing customization is critical to success of it and especially with public cloud based ERP it becomes more challenging. [1] [3] [6] [9].

DOI: 10.17148/IJARCCE.2023.12668

II. LITERATURE REVIEW

In their research, Moutaz Haddara et al. (2022) delved into the challenges faced by Small- and Medium-sized Enterprises (SMEs) when adopting cloud-based enterprise resource planning (ERP) systems, as discussed in the literature. While cloud-ERP systems offer new opportunities for companies, including SMEs, to access and benefit from ERP functionality, the adoption and implementation of these systems still pose challenges for organizations venturing into cloud-based software and new systems.

One significant barrier highlighted on Page 976 is customization, which SMEs often encounter when adopting and implementing cloud ERP solutions. Cloud-ERP packages typically come in standardized forms, making customization a difficult, costly, and time-consuming process. Therefore, the flexibility of both the SME and the cloud service vendor becomes crucial in the ERP adoption journey [8-10, 33]. A case study conducted by [11] revealed a divergence in SMEs' perceptions of Software as a Service (SaaS) ERP. While two participating companies considered it a permanent solution, two others viewed it as an interim solution before eventually transitioning to a customized on-premises ERP system.

The limitations of customizing cloud-ERP systems can deter SMEs from implementing such solutions, particularly if they anticipate future expansion that necessitates more complex ERP operations. Negligence in selecting a suitable cloud-ERP package can significantly impact the adoption process, affecting timeframes, budget allocation, goals, and overall success. It is crucial for SMEs to carefully consider their customization needs and evaluate the compatibility of cloud-ERP systems with their expansion plans and long-term business requirements.

WiamYaheaet. al. (2020) in their study compares on premise ERP and cloud ERP solution and acknowledges customizing as major challenge especially in case of cloud based ERPS. It’s mentioned that “Cloud ERP, customization is not an easy task and could be one of the significant reasons for the delay in the project implementation, especially when the vendor does not follow the best practice while customizing the application. On-premise ERP provides exceptional ability of customization. ERP companies put more restrictions on the cloud ERP customization to maintain their ERP system's unified identity. Also, they force partners and customers to have a test environment where any customization should be published and tested on it; then it can be published on the production environment after making sure that there is no conflict with the original ERP code and functions. Core ERP business logic doesn't allow partners to customize or extend it to ensure the system will keep working as expected. They provide what we called models and data models, where any customization can be maintained in these models and can be removed at any time without affecting original ERP system functions and features.”

In their study, Qian Huang et al. (2021) highlighted the significance of Cloud-based Enterprise Resource Planning (CERP) systems as a strategic tool in the modern competitive business landscape. However, implementing these systems poses challenges due to the need to address various risks to ensure successful implementation. One notable challenge is customization, which has received limited research attention from Information Systems (IS) researchers despite its importance in CERP implementation.

The primary focus of their study was to shed light on two key aspects: a) the influence of key stakeholders' engagement and interaction on the customization processes of CERP systems, and b) how the actual benefits resulting from CERP customization are realized. By exploring these factors, the researchers aimed to provide insights into the intricate dynamics and outcomes associated with customization in CERP implementation.

This study underscores the need for a deeper understanding of customization challenges and its impact on CERP systems. By uncovering the influence of stakeholders and elucidating the tangible benefits derived from customization, the research contributes to bridging the gap in knowledge regarding this critical aspect of CERP implementation.

Phu Hong Nguyen et. al. (2019) has discussed possibility of using micro-service for non-intrusive customization technique. They quote that “Enterprise software vendors often need to support their customer companies to customize the enterprise software products deployed on-premises of customers. But when software vendors are migrating their products to cloud-based Software-as-a-Service (SaaS), deep customization that used to be done on-premises is not applicable to the cloud-based multi-tenant context in which all tenants share the same SaaS. Enabling tenant-specific customization in cloud-basement-tenant SaaS requires a novel approach. Non-intrusive deep customization means that the customization of each tenant is isolated from the main software product and other micro-services for customization of other tenants.”

DOI: 10.17148/IJARCCE.2023.12668

III. NEED FOR CUSTOMIZATION IN CLOUD ERP

ERP software is designed and developed to meet the standard needs of a diverse range of businesses. But every company has unique business requirements that may not be covered by an out of box vendor provided ERP solution. Thus, customization is necessary to ensure that the ERP system caters to company's specific processes and business objectives. Though, Business process standardization can help to certain extent. However, there is lack of universally adopted business process standards for every business process. Thus most of times it’s expected that ERP software will provide capability to be customized at certain extent.

Furthermore, Following are a few reasons why a need arises to customize their ERP system:

➢ Variation/Gap in business processes: Business process differs from industry to industry and organization to organization. Typically, an ERP vendor develops an ERP product based on certain business practice adopted industry wide and allows variation to be met using customization. Thus, a vendor provided ERP by design needs some level of changes by the organization to meet the exact demand of the organization. The changes depend upon how big the requirement gap is between vendor delivered ERP product and the needs to the organization.

➢ Integration& Data visibility needs: An ERP system may need to be integrated with other systems, e.g. an organization might be using a custom data lake or some legacy system or some propriety software’s. So, the ERP system would need to have provision to interact with third party systems as well as a way to analyze & present the data in certain way.

IV. CUSTOMIZATION TECHNIQUE IN CLOUD ERP

There are various ways to handle the customization need at various aspect of ERP software. Mostly customization involves low code or no code at UI, Business logic layer & data base layer level.

➢ Configuration Options: SaaS software offers various settings that can be adjusted to match specific requirements. This includes customizing interfaces, workflows, data fields, and business rules.

➢ Role-based Access Control: SaaS applications allow organizations to define user roles and permissions, ensuring controlled access to features and data based on individual responsibilities.

➢ Branding and Theming: Organizations can customize the visual appearance of the software by incorporating their own logos, colors, and interface designs.

➢ Integration and APIs: SaaS applications provide APIs that enable integration with external systems and the development of custom extensions, allowing for seamless data exchange and automation.

➢ Custom Workflows and Business Logic: Some SaaS software permits the creation of tailored workflows and business rules to align with specific organizational processes and requirements.

➢ Custom Reporting and Analytics: Organizations can generate customized reports, dashboards, and analytics models within SaaS applications to gain insights specific to their needs.

➢ Data and Field Customization: SaaS software allows the addition of custom data fields or modification of existing ones to capture organization-specific information.

➢ Deep Customization: At times, the intricacies of an organization's business logic may diverge significantly from the existing framework. In such cases, the power of deep customization comes to the forefront, allowing businesses to transcend the limitations of preexisting logic and author completely custom implementation

V. SERVER-LESS RUNTIME & FAAS

Server-less architecture refers to architecture in which developer focus on writing code without any worry of infrastructure beneath.

FaaS - as a Service, is a new and innovative cloud computing model (Based on implementation of Server-less architecture) that offers developers the ability to build and run their services without the need for dedicated infrastructure. It’s an implementation of server-less architecture. It is a unique way of delivering software capabilities in a modular and scalable manner, allowing organizations to concentrate on their business functions instead of managing IT environments. FaaS takes the traditional software development process and turns it on its head by breaking down applications into smaller, independent functions that can be triggered by events or API requests. These functions are then managed and executed by the cloud provider, freeing up valuable time and resources for the organization.

A key advantage of FaaS is its scalability. The cloud provider automatically manages the underlying infrastructure, scaling the system up or down based on demand. This means that organizations do not need to invest in expensive hardware or worry about maintenance and upgrades, allowing them to focus on delivering value to their customers. Another important aspect of FaaS is its pay-per-use model.

Pay-per-use model: Organizations only pay for the resources they use, rather than paying for dedicated infrastructure that may go unused. This flexible pricing model allows organizations to control costs and adjust to changing demands. FaaS also provides organizations with a high degree of agility and flexibility.

Less development time: Functions can be deployed and updated quickly and easily, allowing organizations to rapidly respond to changing business needs. This can lead to faster time-to-market for new products and services and improved competitiveness.

VI. PROPOSED APPROACH FOR DEEP CUSTOMIZATION USING SERVER-LESS RUNTIME

In this paper we have experimented with use of function as a service with a combination of event stream to check the viability of providing deep customization in SaaS application. Though there is alternative design option with complex pricing engines however for purpose of this paper we have implemented same via deep customization.

In the figure 1 we have used a reference scenario “pricing element determination” based on a material, customer and sales organization. Based on the user logged in it would determine if there is custom end point and function image. System will call the implementation and displays the determined pricing element to the user.

For this purpose we used “Open FaaS” with FaaS provider as kubernetes and a custom function store. Further Kafka is used to invoke custom functions using Kafka connector. Depending upon if end point is defined in application then the application can either create a topic (alternatively in Kafkaauto.create.topics.enable to true to allow creation of topic automatically). As an implementation partner it’s required to build the function implementation and deploy the image and configure the end point in application.

Thus this mechanism will allow to call custom function exist using /system/function and call the unique implementation using corresponding Docker image. By means of “FaaS Provider” feature it allows to integrating any backend by simply providing the end points and implementing the HTTP handler functions. Thus an ERP vendor can choose to replace kubernetes with any in house platform.

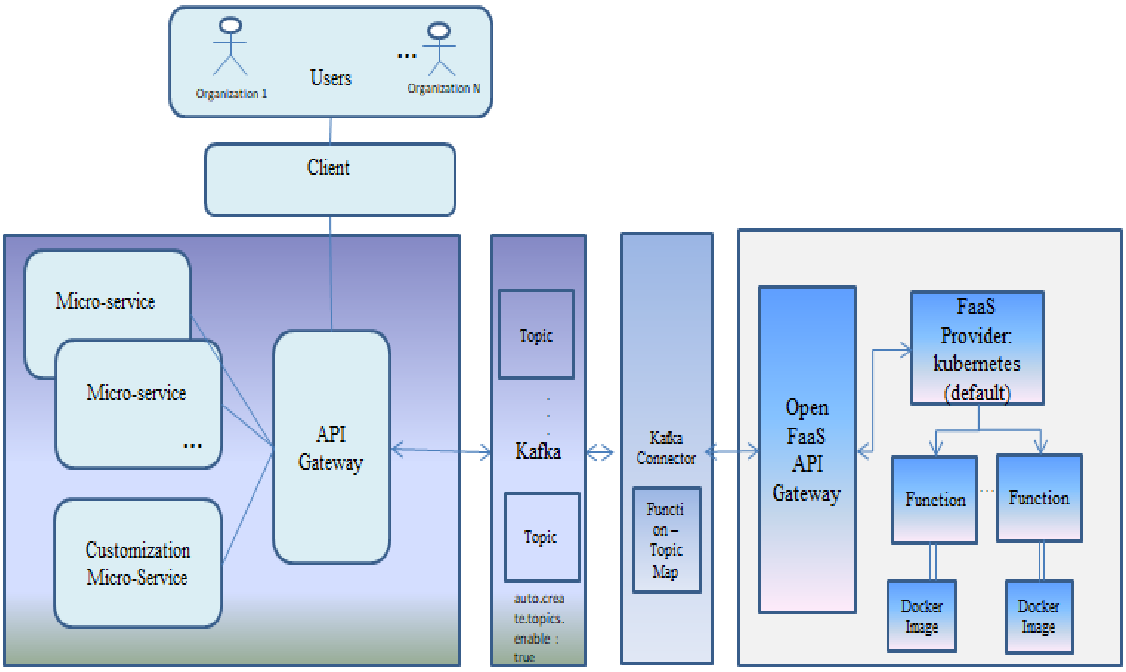

Fig. 1. : Framework for deep customization using FaaS

DOI: 10.17148/IJARCCE.2023.12668

VII. RESULT & FINDING

In this section we show the result during proof of concept for the reference scenario “Pricing element determination”.
1. Above architecture provides a way without any to enable deep customization without “side effect” to the core application.
2. A metering service can be easily integrated to track the usage easily. Thus this approach is a fit for the cloud pricing models as well.
3. This architecture is scalable. Using the containerization, depending upon usage additional pods can be deployed automatically.
4. This architecture is flexible to the extent that especially ERP vendors who are also into PaaS can have multiple choices of replacing the different component with component of their choice e.g. different backend using “FaaS provider”. With fewer changes an API end point configuration can be provided which a user needs to customize at the time of subscribing the application rather than dynamic topic creation.

VIII. CONCLUSION & FUTURE WORK

In this paper, it’s concluded that with careful considerations FaaS can be used to provide capability of deep customization in SaaS ERP software. However, due to limitation of existing implementation of server-less architecture which doesn’t focus on deep customization. In future we need to further refine the architectural components to cater to enterprise needs such as providing lightweight container runtime, lightweight event stream as Kafka is too bulky etc. which are specifically tweaked for need of deep customization.

ACKNOWLEDGEMENTS

I extend my heartfelt appreciation to the Department of Computer Science at Jagannath University, Jaipur, Rajasthan, India, for providing me with the necessary research facilities and for the faculty who have been a constant source of inspiration. I would like to sincerely thank my supervisor, **Mr. Vijay Mohan Shrimal** and **Mr. Suraj Yadav** for their invaluable guidance and support throughout the entire process of preparing this paper. Additionally, I would like to express my gratitude to my parents, **Vimla Devi** and **Shri Shyam Sunder Sharma**, as well as my wife, **Ms. Neeti Sharma** M.Tech. (CS), for their unwavering encouragement and support during my work.

REFERENCES

[1]. MoutazHaddara, Sara Gøthesen and Marius Langseth (2022), “Challenges of Cloud-ERP Adoptions in SMEs”, Procedia Computer Science 196(1) 973–981
[2]. WiamYaheaAlhayek and Rashaahmad Abu Odeh (December 2020), ”Cloud ERP VS On-Premise ERP”, International Journal of Applied Science and Technology Vol. 10, No. 4.
[3]. Qian Huang, Gillian C. Oliver, MD Mahbubur, misitaanwar and Susan Foster (2021), “Understanding cloud-based ERP customization from key stakeholder’s perspective”, Twenty-Ninth European Conference on Information Systems, [Marrakesh, Morocco|A Virtual AIS Conference]
[4]. Phu Hong Nguyen, Hui Song and Franck Chauvel (Aug 2019), “Using Microservices for Non-intrusive Customization of Multi-tenant SaaS”, he 27th ACM Joint European Software Engineering Conference and Symposium on the Foundations of Software Engineering (ESEC/FSE ’19) At: Tallinn, Estonia
[5]. Peng, G.C. and Gala, C.J (2014), “Cloud ERP: “a new dilemma to modern organisations? Journal of Computer Information Systems”, Journal of Computer Information Systems, 54 (3).
[6]. Wei-Tek Tsai* and Xin Sun (March, 2013), ”SaaS Multi-Tenant Application Customization Conference Study” IEEE Seventh International Symposium on Service-Oriented System Engineering
[7]. Mohamed A.AbdElmonema, EmanS.Nasr and MervatH.Geitha (Feb, 2017), “Benefits and challenges of cloud ERP systems - A systematic literature review”, Future Computing and Informatics Journal Volume 1, Issues 1–2, December 2016, Pages 1-9
[8]. ElinUppström, Carl-Mikael Lönn, Madeleine Hoffsten and JoakimThorström (2015) “New Implications for Customization of ERP Systems”, 48th Hawaii International Conference on System Sciences
[9]. Shivam Gupta, Subhas C. Misra, Akash Singh, Vinod Kumar and Uma Kumar (2017), “Identification of challenges and their ranking in the implementation of cloud ERP: A comparative study for SMEs and large organizations", International Journal of Quality & Reliability Management”, International Journal of Quality & Reliability Management, Vol. 34 No. 7, 2017 pp. 1056-1072.
[10]. Marko Mijač, Ruben Picek and ZlatkoStapić (July 2013), “Cloud ERP System customization challenges”, Central European Conference on information and intelligent systems, page 132 to 296

DOI: 10.17148/IJARCCE.2023.12668

BIOGRAPHY

Mr. Naveen Trivedi is pursuing his M.Tech. from Jagannath University Jaipur. He has completed his Bachelor of Engineering in Computer Science & Engineering from Govt. Engineering College Bikaner, Rajasthan in 2007. His area of interests lies in the field of Revenue Growth Management, Life Sciences; Cloud computing, Machine Learning & Artificial Intelligence. He has 16 years of industry experience in various roles such as software developer, Architect, Product Manager and Senior Manager. He has been part of development team for many renowned ERP products as well as well large sized ERP implementation at Fortune's top 500 companies.

Mr. Vijay Mohan Shrimal is an assistant professor at Jagannath University, Jaipur. He completed his Polytechnic Diploma in CS&E from Jodhpur Technical Board, Rajasthan in year 2004 and Bachelor of Engineering in Computer Science from University of Rajasthan in year 2007 and Master degree (M. Tech) from Jaipur National University, Jaipur in year 2012. His area of interests lies in the field of Cloud computing, Machine Learning & Artificial Intelligence, Embedded System and Principles of Programming Languages and many other areas. He has 15 years Academic/Research experience. He is a Member of Computer Science of Teachers Association Collaborate with Association of Computing Machinery, International Association of Computer science and Information technology and International Association of Engineers. He has published 4 patents, 3 books and many research papers in National and International Journals.