A framework to Modernize SME Application in Emerging Economies: Microservice Architecture Pattern Approach

Munezero Immaculee Joselyne, Benjamin Kanagwa , Joseph Balikuddembe

Makerere University
munejosy@gmail.com, bkanagwa@gmail.com, jbalikis@gmail.com

Abstract

The dynamic modern economy of today has resulted the increase in amount of information that enterprise can handle, this imply the usual update of the system to handle new features that are emerging every day. These updates increase the complexity of existing system as those systems become inflexible to any change especially in small and medium enterprise. Enterprises which have microservice architecture are more scalable to release new features in production that are emerging every day. However, for SMEs to change their architecture towards microservice architecture, they have challenges on right microservice patterns of integration to use. As any software development project, modernization projects should be planed carefully and have a good methodology to guarantee successful execution and avoid failure that usually come after unplanned modernization. In this paper, we propose a framework for the modernization of SMEs applications in emerging economy by integrating microservice to existing system, this framework is the combination of the best microservice patterns to be used as execution plan for modernization, in order to improving SME application's scalability and maintenance We have illustrated objectives of this modernization and discussed the role and use of each pattern choice in the corresponding objectives.

1. Introduction

Small and Medium Enterprise (SME) have a strong influence on the growth of the developing world economy and a good contribution to overall gross domestic product (GDP. Over the past decade, SME have adopted computer system, but some legacy or monolith systems that SMEs perform, become inflexible to any change due to low scalability of monolithic architecture, thus they lose the ability to keep pace with necessary new technology that emerges every day.

Monolithic system of enterprises has difficulty to updating any change due to the large code base which is difficult to understand and modify. Moreover to change those systems there are risks that SME should take; firstly, engage in modification that involve high costs, lack of knowledge and security implication of the improved system. Secondly, high risk of failure(Sharif Mohamud, 2014)(Shaul & Tauber, 2013). Engaging in modernization requires a clear understanding of the defect of legacy system towards business loss, hence finding an appropriate approach to eradicate the problem(Global, 2016). Enterprise chooses modernization because; they want to improve the performance, fix the existing system flaw, keep pace with new environment, or plan for the future of the business(Scott, 2016). But the main reasons why the company can take a modernized system is that; it should provide what the company needs, insofar as it supports the purpose of the company and succeeds its objectives. In addition, to ensure that this goal must be achieved, the company should make a good choice on which strategy to use that can support the business and increase the company's performance.

Depending on what the business expose to IT, some SME have modernized their legacy system through integration of Service Oriented Architecture (SOA)(Ravi Khadka, Amir Saeidi, Andrei Idu, Jurrian Hage, 2012)(Muhammad & Galvão, 2016) for black box approach. These methods have been used mostly for enterprise that need to add web based services such as RESTfull web service in the system, and increase operation flexibility using web based user interface (Ravi Khadka & al., 2012). However, this approach is most appropriate in situations where the legacy system works well. While web modules improve the applications appearance and provide a false sense of security about improved system functionality, they add to the system complexity and maintenance cost(Daniel et al., 2016). Others opt for complete transformation of existing system which increase the risk of failure

and cost of new system (Cognizant, 2015).

In this paper we propose a framework approach of gradually transforming the legacy system and integrating microservice architecture pattern. Microservice architecture introduces a new application design strategy that uses independent fine grained services to compose an application, compared to monolithic application style(Irakli & al., 2016), Microservices are mainly used by enterprise to deploy large and medium applications as a set of small independent services that can be developed, tested, deployed, scaled, operated and upgraded independently(Patanjali et al., 2015). Moreover, enterprise that use the microservice architecture provide rapid functional changes which contributes to achieving high integrity factors such as maintainability and scalability (Thones, 2015)(Jarman, 2015). Some enterprises have modernize their systems through migration from monolithic to microservice (Amazon, Netflix,..), however because this enterprises are considered as big company, their practice will lack applicability in SME context (Putra & Hasibuan, 2015).

Research available for modernization of SME application by using microservice architecture approach are specific to a particular situation, this imply the lack of the guidance on the best pattern to use , so there is a need to customize this practice to SME context.
This paper address the above issues by providing a framework for modernization of SME application. Our framework follow principles of Architecture Driven Modernization (ADM), to integrate microservice patterns to existing system after restructuring or re-engineering, so that it can interact with new microservice. It concern of modernization by improving existing system for the purpose of functionality reuse, software improvement, modifiability, interoperability and refactoring. The modernization process consists of understanding the architecture and features of existing SME system, restructure existing system, develop microservice, and develop integration process and continuous monitoring activities in Dev/OP microservice strategy.
We choose SME of developing country because their situation and that of developed country are not the same (Odusote & Adigun, 2015) in most of the cases they are not exposing the same resources to the business and IT .

The rest of the paper is organized as follows; Section 2 provide related works of modernization strategy, Section 3 describes our approach to come up with the framework, Section 4 discuss the framework and section 5 concludes the paper.

2. Related work

Modernization of SME application strategy

Enterprise must achieve its objective by selecting different modernization techniques. Features such as time, performance or resources that affect this decision need to come from a business case and the value of the investment needs to be measured. By measuring what the enterprise could get in modernization, different company opt for; Replacement, Re-engineering, Migration, or wrapping strategy for modernization (Almonaies, Cordy, & Dean, 2010), or a mixture of those methods.
Replacement mostly called “big bang” by developers consist of replacing entirely the existing system by a off the shelf product. Re-engineering or re-development consists of the analysis and adjustment of an application in order to represent it in a new form that facilitate the communication with services(Austin, 2012). Migration, consists of moving the entire legacy system and its core framework to the new environment gradually(Galinium & Shahbaz, 2014) another is wrapping approach which consists of providing a new service interface to a legacy components by making it easily accessible by other software components (B. Zhang, Bao, Zhou, Hu, & Chen, 2008), this practice is mostly used in SOA.

Monolithic Architecture for SME legacy system

Legacy system that most of SMEs use has a monolithic architecture, these monolith is known as simple architecture, simplified testing, simple to deploy and test. However there are challenges when you add a new functionality, the entire system is affected because it lacks agility to continuous delivery. Some of SME have experienced challenges of working with monolithic architecture; In eMarketing, it reach lead times for a new release

to be shown in the eMarketing platform (Brüggemann & al., 2012), the current monolithic system of NRDC can't scale and if there in a maintenance the system require complete suspension of all services(Le et al., 2014). These are few examples of the challenges of monolithic architecture, in general monolithic architecture of SME are inflexible to the changes and have the following disadvantages; It becomes difficult to release a new feature easily and frequently, the release planning takes a lot of time for making sure if the application will not break down. As monolithic applications grow in size, the deployment become frustrating and slow due to the tight coupling of components and difficulty to replace a components without affecting the performance of the whole architecture

Microservice

Migration from monolithic architecture explained in II-B to microservice architecture is the common modernization methods that enterprise makes to implement rapid functional change of their system. Thus, monolithic application requires refactoring towards more distributed system of independent services that can be optimized independently. It add in an application the ability to scale, alter, remove, or change the service without affecting entire system(Irakli et al., 2016).

In addition, monolithic system are migrated to the cloud microservice (Sabiri & Benabbou, 2015), and different patterns have been developed for this migration (Guo et al., 2016)(Balalaie, Heydarnoori, & Jamshidi, 2015). The main idea of monolith to microservice migration is to increase scalability, ease system distribution, and create a lightweight independent system that can be deployed automatically. In all cases of monolithic to microservice migration, it is not a straightforward process, it require code modification so that depending on specific objective to achieve, the existing system is redesigned, recoded or re-containerization to the purpose of new architecture(Fanell et al., 2016) Failures that occur after the modernize the existing information system are mainly related to the use of bad approaches that cannot solve the problem. Some approaches such as SOA have been used to modernize SME but the problem exist on its centralized architecture of ESB (Miri, 2017) which can be failure point of entire system.

3. Our Approach

Our approach of designing an SME modernization framework is based on modernization steps such as; Modification of existing system, integration of new microservice as well as continuous monitoring. Through literature all modernization require an execution plan and methodology to follow in order to avoid the failure that can occur during operation of modernized system(Kolici et al., 2013). The framework that we developed follow principles of Architecture Driven Modernization(ADM)(Ulrich, 2011). ADM is concerned of modernization by improving existing system for the purpose of functionality reuse, software improvement, modifiability, interoperability and refactoring.

By putting together the quality attributes of ADM, in this research we mixed re-engineering and migration, so that the system provides new functionality and the valuable information of legacy system should be preserved. This methodology is chosen because it is more adapted in the context of SMEs, because it relies on the progressive integration of the microservice system into the legacy system. Our boundary is the SME systems developed in OOP language design, especially systems developed in JAVA, JS, PHP...because are modifiable compared to others (Redhat, 2016).

Initially, legacy monolithic system have composite design of user interface, business logic and configuration file, this architecture have characteristics that all applications modules, files and configuration are packaged in one enterprise archive(EAR) file. In order to let new microservice interoperate with existing system, this must be restructured by code modification and deploy each module to its own package.

4. SME Modernization Framework

This research framework presented in figure 2 show different activities that should be conducted to meet modernization objectives; Understanding current system, restructure existing system. Develop microservice. Develop integration, continuous monitoring in Dev/OP microservice strategy. Within each objective, there are works and process to be conducted explained as follows:

Understanding current system: The process of understanding existing system involve; acquire knowledge on the domain and the actual system architecture through documentation or stakeholder. These processes are followed by reverse engineering which will reveal the working functionality as well as dead code that can be removed.

Restructure existing system: Redesign monolithic system is a difficult process because initially legacy code lack flexibility, thus it is trivial to analyze the code as there is no documentation of the design because original design have been changed during years of operation. The first stage for restructuring existing system is to change the packaging by splitting the one archive file that contain all applications in SME to individual WAR file, and deploy each module in independent server such as Docker container. This process will involve code modification to refine business logic to each module. Furthermore, continuous delivery pipeline is the strategy for delivering quality software through process of testing, verifying and validating new business capability deployed. Our continuous delivery pipeline at this stage will allow performing component testing of module that has been re-factored.

Develop microservice: The development of a new microservice is carried out using the principle of one responsibility at time following the bounded context of the domain. Therefore deploy each service in its container through the same continuous delivery pipeline. Then, create API gateway that will isolate the client and the back end. This API may be in different types depending on the client platform to accessing the system, which is done by running a client specific adapter that can choose each client a suitable API for its requirement. Thus, REST/HTTP API can be chosen for web application and mobile application client, where gRPC it is for 3rd party application. Moreover each backend service exposes a REST API and most services consume APIs provided by other services or end users. The API should expose security to the system. The user will pass an access token containing its information to be authorized to access the system. At this stage unit testing must performed to verify whether microservice is performing only one responsibility of the domain at time.

Develop integration: Depending on the nature of communication, integration of Re-factored SME application to new microservice is done on API gateway as synchronous communication using HTTP protocol, to connect end user to microservice. Furthermore, service to service, the communication is asynchronous using pattern like AMQP,or Akka. In microservice based architecture, the service are communicated through a lightweight communication; When there is a request a microservice publish an event which other party can consume to complete the process request. Hence pattern of service discovery (Eureka,) and load balance are required for that action. During Integration, modules for existing system are integrated with new microservice by using REST API gateway. In addition microservices are designed to cope with failure, based on the nature of communication of microservices, it's quite possible that a service could fail, for one reason or another. In these instances, another service can continue functioning while the failed service is adjusting itself, this process is known as circuit breaker. Service breaker concern of detecting the failures quickly and, if possible, automatically restore service.

Dev/Op: Microservices architecture is an evolutionary design and, this design require a group of application developer to monitor changes during operation of new system. Services in this architecture are communicated through network protocol, the particularity of this architecture on SME is the usage of intranet connection to pull information from multiple service or database from one part to another. The intranet connection will make the services available and help SME to effectively update a database even if there is no Internet connection. Moreover software systems relies on hardware resources.

Data: For each step of modernization, data have to cope with the rest of architecture. Microservices architecture pattern significantly impacts the relationship between the application and the database because data have to be decentralized. Initially data of existing SME system are located in one relational database. The refactoring of data from relation database to no relation database (preferred with microservice) will depend on the nature of data which is stored (blog data, flat data). The aim of refactoring data is to insure the dependency of service, so that each module can have a dependent database Data in this architecture must be integrated then make usual update to warranty the consistency in database.

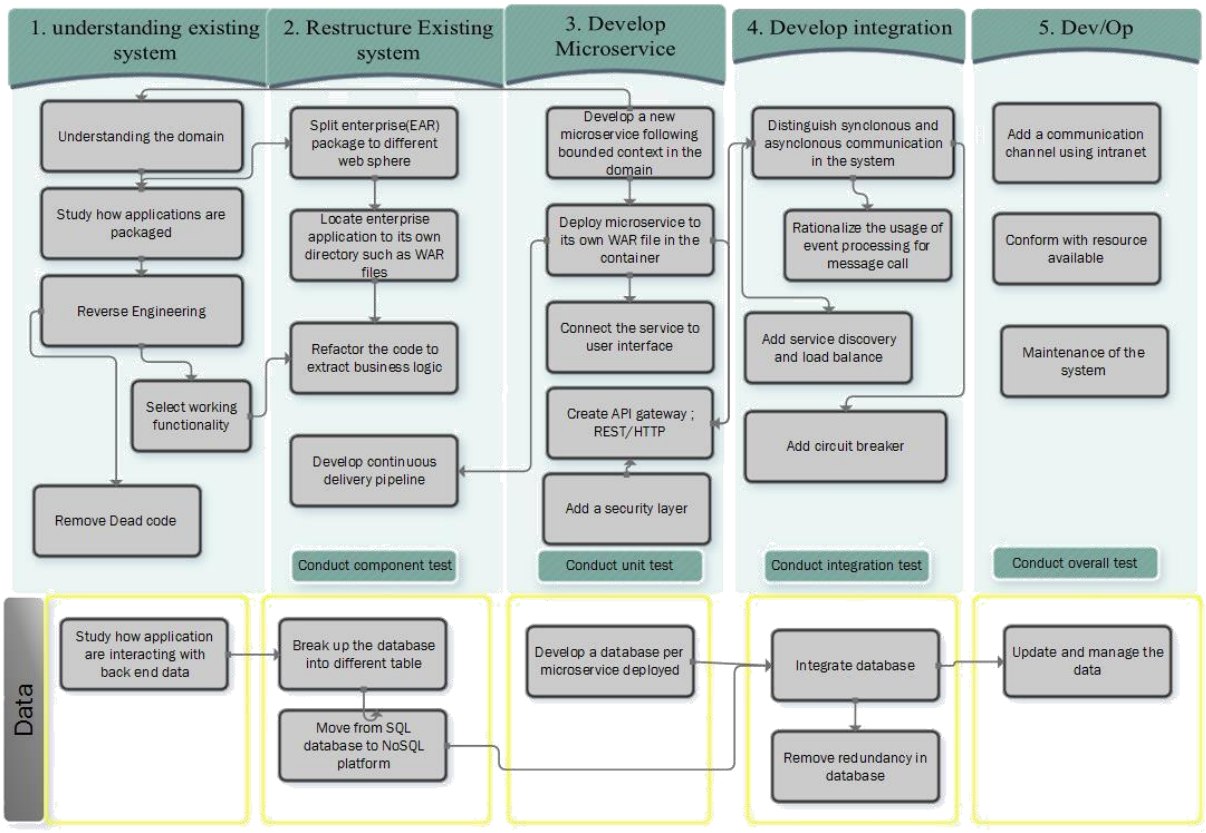

Fig 2: SME application modernization framework

5. Discussion

A framework of modernize SME application using microservice approach presented in section IV is designed to illustrate activities to be conducted towards a modernization of SME application in the emerging economy. This framework is a good choice for integrating gradually microservice to SME by achieving two goals: The first goal of this modernization is the reuse of existing functionality. Therefore With understanding of existing system we seek to retrieve its valuable information that can be used for next stage of modernization. Since microservice has to interact with monolithic system, it requires compatibility criteria that will enable SME system interoperate with new microservice, which is why monolithic system has to be refactored to separate its components so as each module can be deployed in its own package.

The second goal of this modernization is to make a modernize system accommodate new functionality through deployment of new service on continuous deployment pipeline. Thus any new feature that must be added to the system will require the deployment of new microservice.

Microservice architecture is widely used in cloud computing, which involve the usage of internet in the architecture. The particularity of this framework in SME of emerging economy is the use of intranet to cover low latency of internet in SME context. The use of intranet is necessary because microservice communicates through network protocols. Therefore coexisting SME applications and microservice require continuous monitoring of communication path by the production team.

Conclusion

In this paper we presented a framework for SME modernization by using microservice architecture, this framework involve effective usage of patterns that will make SME application and microservice coexist in production. Modernization of enterprise system requires planning to avoid failure that can occur after modernization. In this paper we show how SME of emerging economy have challenge of right modernization technology choices on microservice architecture that can solve their problems without relying on big enterprises. To solve this challenge we designed a framework that show how right patterns of modernization can be used in SME context. However in the future work, we envisage expanding and using this framework in practical case.

References

Almonaies, A. A., Cordy, J. R., & Dean, T. R. (2010). Legacy System Evolution towards Service-Oriented Architecture. *IEEE*, 1–8.

Austin, P. F. (2012). Applying system engineering processes to legacy test program set modernization. *AUTOTESTCON (Proceedings)*, 70–74. http://doi.org/10.1109/AUTEST.2012.6334541

Bajahzar, A., Alqahtani, A., & Baslem, A. (2012). A Survey Study of the Enterprise Resource Planning System. *2012 International Conference on Advanced Computer Science Applications and Technologies (ACSAT)*, 246–252. http://doi.org/10.1109/ACSAT.2012.101

Balalaie, A., Heydarnoori, A., & Jamshidi, P. (2015). *Microservices Migration Patterns*.

Brüggemann, M. E., Vallon, R., Parlak, A., & Grechenig, T. (2012). Modelling Microservices in Email-marketing Concepts , Implementation and Experiences THE CONCEPTS OF EMAIL MARKETING AND. Research Group for Industrial Software, Vienna University of Technology.

Charette, R. N. (2016). Dragging Government Legacy Systems Out of the Shadows. *IEEE Computer Society*.

Cognizant 20-20 Insights. (2015). Legacy Enterprise Systems Modernization : Five Ways of Responding to Market Forces. Cognizant 20-20 Insights.

Daniel, E., Cardenas, D., Rolando, A., Castro Eddie, Garc, K., Parra, C., & Casallas, R. (2016). Towards the Understanding and Evolution of Monolithic Applications as Microservices. *IEEE*.

Fanelli, T. C., Simons, S. C., & Banerjee, S. (2016). A Systematic Framework for Modernizing Legacy Application Systems. In *2016 IEEE 23rd International Conference on Software Analysis, Evolution and Reengineering*. http://doi.org/10.1109/SANER.2016.40

Galinium, M., & Shahbaz, N. (2014). Case Studies: Business and Technical Perspectives in Migration of Legacy Systems to Service Oriented Architecture. *ECTI Transactions on Computer and Information Technology*, *7*(2). Retrieved from http://arxiv.org/abs/1412.7959

GLOBAL, U. (2016). Legacy modernization. UST GLOBAL |.

Guo, D., Wang, W., Zeng, G., & Wei, Z. (2016). Microservices Architecture based Cloudware Deployment Platform for Service Computing. *IEEE Computer Society*, 358–364. http://doi.org/10.1109/SOSE.2016.22

Irakli, N., Ronnie, M., Matt, M., & Mike, A. (2016). *Microservice Architecture*. (B. M. and H. Bauer, Ed.) (1st editio). O'Reilly Media Inc.

Jan, D. (2014). *Information Systems for Small and Medium-Sized*. (H. van L. Jan Devos, Ed.) (1st editio). Springer-Verlag Berlin Heidelberg, Dirk Deschoolmeester.

Jarman, P. (2015). Microservices – A New Application Paradigm. *Infosys*.

Kolici, V., Xhafa, F., Pinedo, E. E. D., Nunez, J. L., Segui, V., & Barolli, L. (2013). Analysis of Mobile and Web Applications in Small and Medium Size Enterprises. In *2013 Eighth International Conference on P2P, Parallel, Grid, Cloud and Internet Computing* (pp. 324–330). http://doi.org/10.1109/3PGCIC.2013.55

Le, V. D., Neff, M. M., Stewart, R. V, Kelley, R., Fritzinger, E., Dascalu, S. M., … Microservice, K. (2014). Microservice-based Architecture for the NRDC. *IEEE*.

Mahfudhi Muhammad, G., & Galvão, D. T. (2016). Specifying Modernization into Service-Oriented SaaS System in a Case of Public Transport Document Generator. *Springer International Publishing*, 590–603. http://doi.org/10.1007/978-3-319-32689-4

Miri, I. (2017). Microservices vs. the monolith. Retrieved June 6, 2017, from https://dzone.com/articles/microservices-vs-soa-2

Odusote, B. O., & Adigun, M. O. (2015). Technologically enabling resource-constrained enterprises in developing nations through the implementation of E-Commerce on-demand service portals for a grid utility computing platform. *IEEE International Conference on Adaptive Science and Technology, ICAST*, *2015-Janua*. http://doi.org/10.1109/ICASTECH.2014.7068149

Patanjali, S., Truninger, B., Harsh, P., & Bohnert, T. M. (2015). CYCLOPS : A Micro Service based approach for dynamic Rating , Charging & Billing for cloud. http://doi.org/10.1109/ConTEL.2015.7231226

Putra, P. O. H., & Hasibuan, Z. A. (2015). E-business framework for small and medium enterprises: A critical review. In *2015 3rd International Conference on Information and Communication Technology, ICoICT 2015* (pp. 516–521). http://doi.org/10.1109/ICoICT.2015.7231478

Ravi Khadka, Amir Saeidi, Andrei Idu, Jurrian Hage, S. J. (2012). *Legacy to SOA Evolution : A Systematic Literature Review Ravi Khadka*.

Redhat. (2016). *A PLATFORM FOR MODERNIZING*.

Sabiri, K., & Benabbou, F. (2015). Towards a Cloud Migration Framework. *IEEE*.

Scott, T. (2016). *LEGACY SYSTEM MODERNIZATION Addressing Challenges on the Path to Success . ACT-IAC.*
Sharif Abdullahi Mohamud, S. B. (2014). Risk Assessment Factors for SME Software Development Companies in Malaysia. *Ieee*. http://doi.org/978-1-4799-0059-6/13/$31.00
Shaul, L., & Tauber, D. (2013). Critical success factors in enterprise resource planning systems. In *ACM Computing Surveys (CSUR)* (Vol. 45, pp. 1–39). http://doi.org/10.1145/2501654.2501669
Thones, J. (2015). Microservices. *IEEE Software*, *32*(1), 116–116. http://doi.org/10.1109/MS.2015.11
Ulrich, W. M. (2011). Architecture - Driven Modernization 101: Concepts, Strategies & Justification. Tactical strategy Group Inc. Retrieved from www.systemtransformation.com
Zhang, B., Bao, L., Zhou, R., Hu, S., & Chen, P. (2008). A Black-Box Strategy to Migrate GUI-Based Legacy Systems to Web Services. *IEEE International Symposium on Service Oriented System Engineering*, 25–31. http://doi.org/10.1109/SOSE.2008.8
Zhang, Y. (2010). An analysis on the ERP application status of the small and medium manufacturing enterprises. In *2010 2nd International Conference on Communication Systems, Networks and Applications, ICCSNA 2010* (Vol. 2, pp. 165–168). http://doi.org/10.1109/ICCSNA.2010.5588901