SERVERLESS COMPUTING: A MULTIVOCAL LITERATURE REVIEW

Mubashra Sadaqat
Østfold University College
Halden 1783, Norway
mubashra.sadaqat@hiof.no
Ricardo Colomo-Palacios
Østfold University College
Halden 1783, Norway
ricardo.colomo-palacios@hiof.no
Lars Emil Skrimstad Knudsen
Østfold University College
Halden 1783, Norway
lars.e.knudsen@hiof.no

Abstract

Serverless computing is a cloud computing execution model which enables developers to focus more on business logic rather than on infrastructure or maintenance of servers. This new paradigm has become a source of attraction for developers and organizations alike as it does not only reduce but simply eliminates the overhead of scaling, provisioning and infrastructure altogether. Given the novelty of the phenomenon, this paper is meant to study the phenomenon in a systematic way in order to define the core components of serverless computing, its benefits, challenges and what lies in the foreseen future of the serverless concept. To this end, authors conducted a multivocal literature review in order to better comprehend the state-of-art on serverless computing. The study shows that serverless computing is a solution that allows users to create functions that intercept and operate on data flows in a scalable manner without the need to manage a server, although presents several challenges.

1. INTRODUCTION

Serverless computing is rapidly gaining attention from IT practitioners and academics alike. Serverless computing is an emerging cloud computing paradigm that provides a platform to efficiently develop and deploy applications to the market without having to manage any underling infrastructure (Lloyd et al. 2017). Taking into account this lack of infrastructure, and according to Rai et al (2017), serverless computing is much closer to the business aspects rather than the technical ones.

To a certain extent, the term serverless is confusing. If we look at the word "serverless", it's easy to assume that it refers to computation which excludes servers entirely (Robert 2016). However, it is not technically possible to serve the requests of clients without servers (Rai et al. 2017). As wireless internet has wires somewhere, likewise serverless architectures will still have servers somewhere (serverless 2018). The term "serverless" is used in the context of development which transforms servers and their management into a level of abstraction for developers and shifts all the underlying infrastructure workload towards cloud vendors (Adzic et al. 2017, Yan et al. 2016, Wolf 2017). The serverless computing platform facilitates and enables developers to focus only on business logic, without the overhead of scaling, provisioning and infrastructure as the program technically runs on external servers with the support of cloud service providers (Baldini et al. 2017, Yan et al. 2016).

In recent years, we have seen an incredible evolution of servers; from physical servers to virtual servers in data centers, and from virtual servers in data centres to virtual servers in the cloud, where all traces of actual physical servers has vanished (Barga 2018, Persson et al. 2017). In the traditional architecture, developers were required to work on provisioning, managing, and maintenance of servers, which is both time-consuming and

result in higher costs (Barik et al. 2018). Moreover, it significantly distracts developers from what should be their core focus by making them spend countless hours on server management (Sbarski 2017). Alternatively, in serverless computing, developers need not be concerned with provisioning, monitoring, maintenance, scalability, and fault-tolerance (Malawski et al. 2017). Instead, they are able to utilize the acquired time for the development of application and their core business logic. (Glikson et al. 2017).
Back in 2006, Amazon launched a cloud storage service: AWS S3. It provided a storage service without the need of handling the maintenance of the servers (Adzic et al. 2017). Later on, in late 2014, a serverless platform was launched by Amazon Web Services in which computation service became serverless. And it was from this point onwards that the new paradigm of cloud revolution began (Eivy 2017). Not so long after, in 2016, other major cloud vendors released serverless platforms to compete with already existing Amazon Web Services such as Google, Microsoft and IBM (Baldini et al. 2017, C. Wolf 2017). Almost instantly, serverless computing start gaining popularity with organizations opting for the new technology to transform their businesses (Kiriaty 2016). Currently, various commercial and open source serverless services are made available such as AWS Lambda, Google cloud functions, Microsoft Azure functions, IBM cloud functions (Baldini et al. 2017,Persson et al. 2017) and OpenLambda is an open source initiative to provide serverless (Varghese & Buyya, 2018). Subsequently, providers are working progressively to provide the best functionality and services to differentiate themselves from their competitors (Eivy 2017). However, the market is mainly captured by the Amazon Web Services as they already had a strong foothold before the competitors entered the industry (Crane et al. 2017). According to Markets and Markets report, the serverless market will grow from a $1.88 billion market in 2016 to $7.72 billion by 2021 and 80 percent by 2020 (Leitersdorf et al. 2017,Contino 2017).

By making use of serverless computing, companies save significant time, money and resources by only looking out for hosting, running and managing core functionality of applications (Lloyd et al. 2017). The key insight of serverless computing model is to quickly scale up and down without the need to process data in the storage servers (Sampé et al. 2017). Companies using serverless computing platforms are increasingly finding improved business agility or cost savings by renting software to vendors such as Amazon, Microsoft, Google or IBM. Apart from that, and initially, developers are explicitly spared from any worry regarding servers or virtual machines, load balancing across multiple server instances, scaling up and down etc (Crane 2017).

In business terms, serverless computing is used to build and run applications with the support of third party management which executes the relevant code whenever the corresponding events occur and user is required to pay for the duration only when a server is in actual use (Baker et al. 2017). Several companies are using serverless computing such as Netflix, Coursera, Coca Cola, Quest, Unilever along with many more (Boulton 2018, Rapyder 2018).

Serverless computing is a lightweight computing solution that allows users to create small, stateless functions that intercept and operate on data flows in a scalable manner without the need to manage a server or a runtime environment (Sampé et al. 2017). It is a new model of execution that is currently emerging to transform the design and development of modern scalable applications (Pérez et al. 2018). Developers are provided with various levels of abstraction of services by the cloud vendors through which they can also execute their code directly (Chang et al. 2017). However, literature has also reported several challenges on the use of serverless computing.

“Serverless computing” is a misleading, new term and the marketplace of serverless computing platforms has started enjoying immense popularity, with constantly increasing adoption by the IT and business industries. Therefore, it becomes necessary to conduct a detailed study on this topic. Presently, academic literature has not studied the phenomenon in deep and most of the information available is present in grey literature. Moreover, we found that no multivocal literature review (MLR) is yet published on the topic. This led us to conduct our study using the MLR research methodology. By these means, authors will be able to examine the concept in a systematic manner, including both reported benefits, but also challenges.

The paper is organized as follows. Section 2 describes the methodology adopted including research questions. Then, Section 3 presents the search results of the study followed by Section 4 which answers research questions. Finally, Section 5 summarizes the result and concludes the paper.

2. RESEARCH METHODOLOGY

In this section, an overview of the research methodology based on MLR is presented detailing also research questions and data collection.

2.1 MLR

The academic literature available on serverless computing is in quite a limited quantity. Moreover, we found no MLR up till now published on serverless computing. Therefore, we decided to conduct our study using MLR. A MLR is a type of systematic literature review which includes all accessible knowledge on a topic in the form of grey literature such as white papers, web pages, blogs and videos with the combination of academic literature (Garousi et al. 2016, Garousi et al. 2017).

Ogawa et al (1991) mentioned the importance of MLR by stating that by reviewing the published and non-published literature of various authors, adds to the versatility in the understanding of the topic. Furthermore, MLR diminish the gap between academic research and professional practice and it is valuable in order to draw conclusions and further questions for advance research (Elmore 1991). So far the number of papers that has been published using MLR is few and even though grey literature is also included in systematic literature review papers, it does not make use of the term MLR explicitly (Garousi et al. 2016).

MLR are gaining interest in academic literature to combine the study with knowledge of state of arts and practice (Garousi et al. 2017) and as a consequence of this, literature witness several MLR in fields like software testing automation (Garousi et al. 2016), DevSecOps (Myrbakken et al. 2017), software test maturity assessment (Garousi et al. 2017) or serious games for software process standards education (Calderón et al. 2018), naming just the most relevant and recent cases in the broad field of IT.

2.2 Research questions

The MLR is conducted to collect all information based on the state-of-art of serverless computing. Based on the above goal, there are four questions as follows:

- RQ 1: What are the core technological components of serverless computing?
- RQ 2: What are the key benefits of serverless computing?
  - RQ 2.1: What are the key benefits for developers of using serverless computing?
  - RQ 2.2: What are the key operational benefits of using serverless computing?
- RQ 3: What are the challenges in using serverless computing?
  - RQ 3.1: What are the challenges for developers of using serverless computing?
  - RQ 3.2: What are the operational challenges in using serverless computing?
- RQ 4: What is the expected evolution of serverless computing?

2.3 Search strategy

In this section, the search strategy including literature sources, search string used and search process overall are presented.

2.3.1 *Search data sources*

Data was collected to conduct a literature review on serverless computing by using electronic libraries, Google search and Google scholar. The electronic libraries were used to gather scientific literature through journals, articles and conference papers. Whereas grey literature was obtained using Google search engine which includes cloud vendors' official websites, blogs, whitepapers and other webpages. Towards the end, Google scholar was used to review available academic literature which does not exist in digital libraries such as whitepapers.

Presently, many platforms are available that offer serverless computing services both commercially and open source. The major cloud vendors are Google Cloud, IBM, Microsoft Azure and Amazon Web Services that provide the facility to encapsulate business logic while rest of the work is handled by them. IBM OpenWhisk is an open source platform which established a strong footprint in providing cloud services. Therefore, it becomes necessary to review official website of these cloud vendors in order to study in detail serverless computing platforms and their functionality. By making use of this diversity of literature, the results will give a considerably more refined look at the topic in discussion, which is based on the combination of state-of-art architecture and practices. In the first step, the search source is based upon the following digital libraries:

- Springer Link (http://link.springer.com)
- ACM Digital Library (http://dl.acm.org)
- ScienceDirect (http://www.sciencedirect.com)
- IEEE Xplore Digital Library (http://ieeexplore.ieee.org)

In the next step, the data search resource conduct using cloud vendor's websites including:

- IBM (https://openwhisk.apache.org)
- Google Cloud (https://cloud.google.com/serverlesss/)
- Amazon Web Services (https://aws.amazon.com/serverless/)
- Microsoft azure (https://azure.microsoft.com/en-us/overview/serverless-computing/)

In the last step, in order to investigate grey literature, the following sources were used:

- Google search (http://www.google.com/)
- Google Scholar (http://scholar.google.com/)

2.3.2 *Search term*

The search string is constructed in order to retrieve the most relevant results on serverless computing. This string is a combination of keywords and operators which is used to perform searches in electronic libraries and search engines. First of all, choose terms that are most specifically related to the topic. Then choose terms which can be used to discuss the research questions while in-between operators are included to combine search terms in specific ways to broaden or narrow the results. This is the traditional way to search information on online databases and internet. In MLR, the main search terms used are as follows:

("Serverless computing" OR "Serverless") AND
("definition" OR "platform" OR "components" OR "advantages" OR "challenges" OR "evolution").

2.3.3 *Search process*

The search process contains four steps. The first step is to choose the advanced search option from four digital databases to retrieve the documents that satisfy the search string, then from Google search and Google scholar. Secondly, filter the result using above mention search terms. Next, sort out the results after going through the titles, keywords and abstracts of the documents. Then, filter out the papers which are not related to the topic of "serverless computing". Finally, read the full text in order to conclude results of the aforementioned research questions. Using this approach, irrelevant papers were excluded upon title, then upon abstract, and finally on full text. The main credit was given to the papers that offered some general overview of serverless computing.

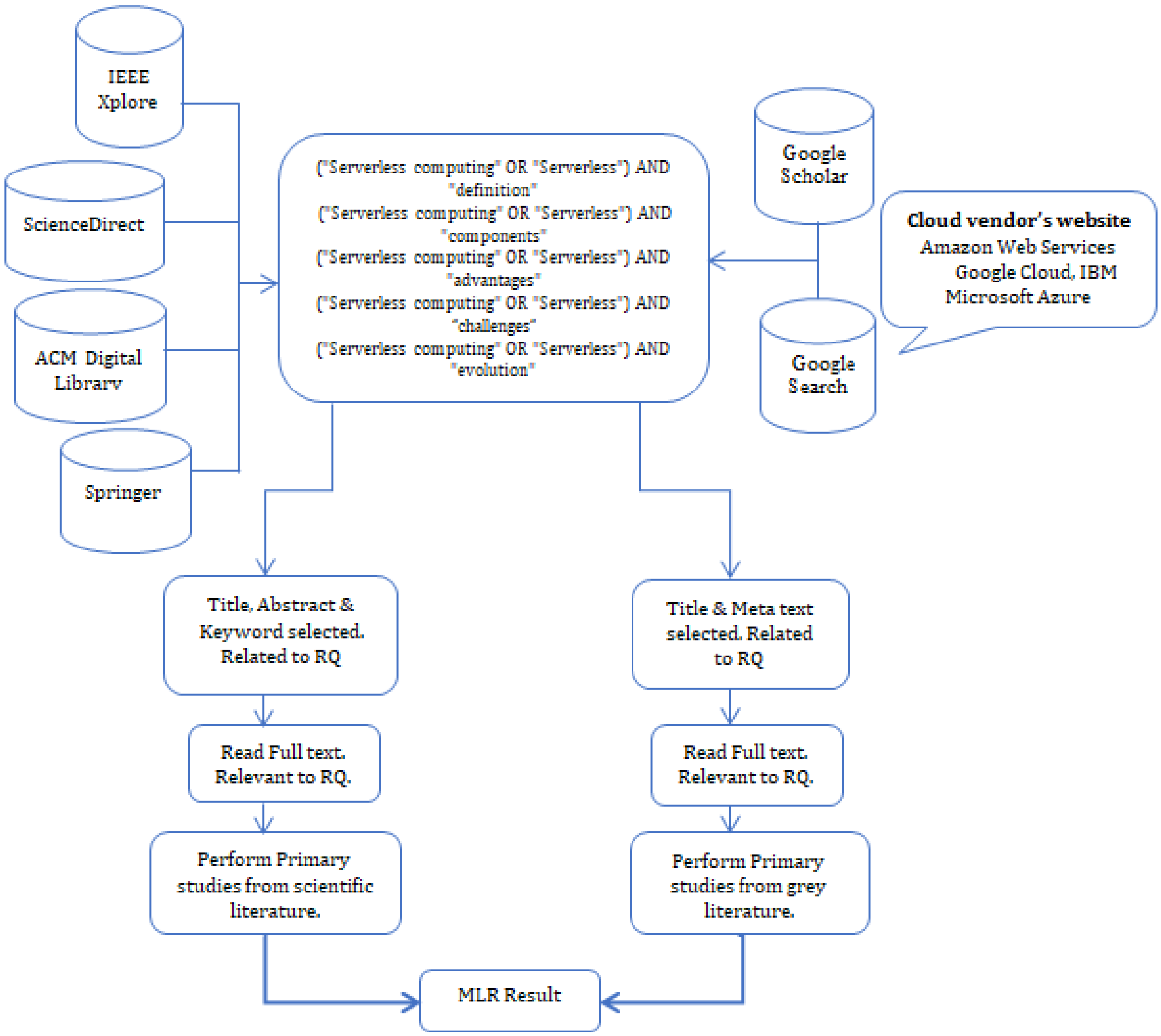

Figure 1. Search process

2.3.4 *Search selection*

The study selection is performed in order to set criteria of inclusion and exclusion which allows efficient filtering of the search results and retrieval of the most relevant literature associated with research questions. This process of inclusion and exclusion establishes the requirements that retrieved papers from the searches must fulfill in order to be included in this study.

- Inclusion criteria:
  - Literature that explicitly discusses serverless computing.
  - Literature that explicitly discusses serverless computing or serverless.
  - Literature that explicitly discusses the components of serverless computing architecture.
  - Literature that discusses the benefit of serverless computing.
  - Literature that explains the challenges in using serverless computing.
  - Literature that predicts the expected evolution of serverless computing.
- Exclusion criteria:
  - Literature available in any language other than English.
  - Literature available explicitly only on computing.

- Exclude duplicate papers based upon title.
- Studies with the full text unavailable.

3. SEARCH EXECUTION

The search procedure was performed using the abovementioned method. Although the initial results reached around 180,000 hits, final set reached 42 documents. Main results of the process are depicted in Table 1 in what follows:

Library	Initial results	Title, Abstract & Keyword selected	Full text
ACM	22	13	7
IEEE	10	7	3
ScienceDirect	5	5	2
Springer	27	3	3
Google Scholar	211	5	1
Google search	181,000	37	26
Results	181,275	70	42

Table 1. search results in the three steps

4. RESULTS

In the following sections, we present the results of the MLR with regards to the RQs. The results of the search process are analyzed taking into account the different primary studies retrieved from both the grey and scientific literatures and the RQs formulated.

4.1 RQ1: What are the core technological components of serverless computing?

The three main technological components of serverless computing are as follows:

Application program interface (API): The API acts as a middleware between the user requests and the serverless platform (Rai et al. 2017). It is the entry point which is used to access data and implement business logic or functionality (Azure 2018 ,Baker et al. 2017).The cloud service providers offer a fully managed service that makes it easier for developers to create, publish, maintain, monitor and secure APIs at any scale (Amazon 2018 , IBM 2018). They can handle requests up to hundreds of thousands of concurrent API calls and ensure traffic management, authorization and access control of services (Baldini et al. 2017). Serverless computing expressively reduces time and effort of developers to build APIs (Kanso & Youssef, 2017). This core service is provided by cloud vendors by the name of Amazon API Gateway, API Management, Google cloud by API Gateway along with other open source APIs that now are available as well (IBM 2018,Sampé et al. 2017). Often, API gateways are significantly helpful in achieving the necessary degree of automation (Spillner 2017), linking the concept to the DevOps world (Colomo-Palacios, Fernandes, Soto-Acosta, & Larrucea, 2018)

Function as a Service (FaaS): The FaaS executes business logic or code by using serverless platforms which provide a level of abstraction for developers (Baker et al. 2017). It allows developers to execute code in response to events without paying attention towards the complexity of building and maintaining the infrastructure (Akiwatkar 2017). Thus, developers are purposefully restricted to only concerns regarding single purpose stateless functions which execute on demand, typically through an API (Malawski et al. 2017, Lynn et al. 2017). FaaS encapsulates the entire business logic while completely extracting the server from the developers perspective (Ast et al. 2017). Such service is provided by cloud vendor Amazon as Lambda function and other major vendors like Google, Microsoft and IBM by name of Google Cloud function, Microsoft Azure function and IBM Cloud functions respectively (McGrath et al. 2017). Azure Functions allow users to run small pieces of code or functions in the cloud (Lloyd et al. 2017). Whereas Lambda functions is an event handler to provide glue code when composing services. And Google Cloud Functions provide functionality to run serverless functions. IBM OpenWhisk provides event-based serverless programming with the ability to chain serverless functions to create composite functions (Baldini et al. 2017).

Backend as a Service (BaaS): The BaaS is a service which removes administration overheads from developers' end while enabling users to directly access databases among other services; this is a third party service provided by cloud vendors. BaaS take away backend workload and provides different mechanisms for authorization to secure it (Barik et al., 2018), Robert 2016). For instance, users get admittance to access databases by means of various security points. The list of cloud databases service providers include Firebase, DynamoDB, Azure Cosmos DB, Cloud Datastore and many more (Malawski et al. 2017). These provide a highly available and durable database which scales automatically to handle applications load (Google 2018).

4.2 RQ2: What are the key benefits of serverless computing?

This section provides the key attributes of serverless computing after a complete review of the scientific and gray literature in order to get information from the perspective of the developer and operational ends.

RQ 2.1 Developer perspective: From the viewpoint of developers, reported benefits are as follows:

No server management: The first benefit of serverless computing is clearly reflected in the term itself as it eliminates the workload of servers and provides a level of abstract for developers (Mcdonald 2017). With a serverless approach, the infrastructure of servers is handled by cloud vendors (Malawski et al. 2017, contino, 2018). Thus, it gives liberty to execute application without taking account of servers, virtual machines or the underlying compute resources (Azure 2018, serverless 2018).

Scalability: In serverless computing, application scales automatically according to demand by adjusting the capacity through toggling of the units of consumption (Malawski et al. 2017). So, the DevOps team is not required to write scripts for scaling up or down as it is managed by the cloud vendors (Baldini et al. 2016). They handle quickly the increase in the amount of traffic (serverless 2018) in a transparent way. This considerably reduces the fear of application crashing (Google 2018, Savage 2018).

Reduced cost: Serverless computing also introduces a new billing model, based on usage rather than pre-provisioned capacity (van Eyk et al. 2017). Unlike traditional servers, it significantly lowers the cost in comparison to the price of maintaining the server (Chang et al. 2017). This means that users do not waste money on resources that are idle or underutilized by them (McGrath et al. 2017) as there is no cost to the consumer when the system is idle (Castro et al. 2017).

Focus on business logic: Developers can focus on core business logic instead of worrying about managing and operating servers, either in the cloud or on-premises (Baldini et al. 2016). This lets developers to reclaim time and energy that can be spent on developing great products which are auto-scalable and reliable (AWS 2018 , Dwyer 2016). Unlike traditional architectures, now they have to just concentrate on the state of resources, time of triggers and then just upload the function to the cloud. All other server workload is handled by vendors (Castro et al. 2017).

Faster time to market: Serverless computing shortens the time between having an idea and the actual deployment of application (Baldini et al. 2017). It reduces the time of deployment to the market by shifting a good amount of load toward cloud vendors. So that developers can quickly build applications (Wolf 2018). It lessens time to market by up to two thirds (Goncharov 2017).

RQ 2.2. Operational perspective: From an operational perspective, reported benefits are as follows:

Security: Serverless architecture provides security benefits also handled by cloud vendors. The automatic scaling capabilities of the serverless functions help mitigate risks from DDoS (Distributed Denial of Service) attacks. It also eliminates attacks which target the vulnerabilities of operation system or by installing malicious software on the company servers (Leitersdorf et al. 2017).

Optimal use of resources: Instead of running an application 24/7 on a server, in serverless computing the application is executed on demand and shares resources which results in elimination of the idle usage of resources. The breakdown of functions into a small section of code which runs based only on an event;

enables optimal resource utilization (Kanso et al. 2017). For example if function needs 500MB of RAM to run and it runs for 1 second, resources are utilize for only that 1 second (Stamat 2016, Golden 2018).

Operational management: The serverless platform provides a clear separation between infrastructure services and applications running on top of the platform. It reduces operational management overheads and lets developers write code directly on the vendors serverless platform. Therefore there is no requirement of continuous integration (Maruti 2017, Eivy 2017).

Operational work: Serverless computing cuts off the need for an IT team to manage the expansion and contraction of server capacity as the serverless architecture scales automatically to meet the demand. And in the case when something goes wrong, it is cloud service providers' responsibility to figure out and fix the problem, reducing in-house personnel efforts (Chan 2017, Eivy 2017).

4.3 RQ3: What are the challenges of using serverless computing?

RQ3.1 Developer perspective: From a viewpoint of developers, reported challenges are as follows:

Vendor lock-in: The features of serverless computing differ among platforms as each vendor has their own flavor of integration points, configuration etc. In order to switch vendors, customers will probably need to change their code and their operational tools (Leitersdorf et al. 2017, tata 2017). It creates a lock for switching from one vendor to another, requiring, then, significant time and efforts (van Eyk et al. 2017).

Demand of skilled workers: To work on serverless computing a higher level of developer capabilities are required which are appropriately suited to work with the platform and developers should also have possess adequate knowledge of refactoring functions such as splitting and merging functions etc (Baldini et al. 2017). Serverless computing brings along with it the complexity of splitting a single application into a fabric of services and functions, which significantly increases with the number and variety of services (Rai et al. 2017).

Testing Complexity: Testing serverless deployments differs from traditional deployments and orchestrating integration tests in this environment is not trivial**.** Moreover, performing load testing in serverless settings is challenging, since there is a dependency on externally provided systems for running testing scenarios (Rai et al. 2017).

RQ3.2 Operational perspective: Retrieved challenges in the operational side are depicted in what follows:.

Security: Initially, there are less attack points in serverless scenarios, however, research is needed to understand the new security issues introduced by serverless computing (McGrath et al. 2017, Baldini et al. 2017).

Monitoring: Serverless computing abstracts away the notion of a server from the developer, and pushes operational concerns to the providers. Despite this, it will be important for developers to monitor the deployment of their solutions since the execution of their functions is directly mapped to a cost model based on execution time (Yan et al. 2016). Monitoring, logging and debugging serverless architecture may add more costs (tata 2017).

Loss control over infrastructure: Service providers control the underlying infrastructure, and developers are not able to customize or optimize the infrastructure (tata 2017). Apart from that, developers are not in command on their environments as vendors can remove, update functionalities or change APIs (Yang, 2017).

4.4 RQ4: What is the expected evolution of serverless computing?

Serverless computing will play an integral role in the future of business transformation and IT industry. Authors analyze its growth in what follows in three different aspects:

Adoption of serverless computing by companies: An increasing amount of companies will continue to leverage serverless computing model (Chan 2017). Those companies who specifically want to build event-based applications quickly will adopt serverless computing platform in order to increase efficiency and boost productivity (Watts 2018), literature also linked innovation with serverless adoption (Sbarski 2017). As the tremendous growth and continuous innovation in the public cloud will force more companies into adopting the serverless computing platform, it will lead to a significant increase in the public cloud adoption rate of companies in 2018 and so on (Rapyder 2018).
Market growth: According to analyst Jason McGee of IBM Cloud, their serverless market is predicted to grow from 7-10x by 2021. This statement is supported by a recent Markets and Markets report which predicts that serverless market will grow from a $1.88 billion market in 2016 to $7.72 billion by 2021 and 80 percent by 2020 (Leitersdorf et al. 2017,Contino 2017). Forrester predicts that the biggest three public cloud vendors Amazon Web Services, Google and Microsoft will capture 76% of all cloud platform revenue in 2018 (Carey 2017). According to a recent Gartner research report, the cloud computation will remain a major cause of the shift in IT spending policies of the companies in 2018. The leading American market research firm has also predicted about the public cloud platforms that the fastest growing segment will generate $44 billion in 2018. (Rapyder 2018).
Cloud vendors: Vendors will expand their boundaries and enhance functionality to cover almost all the requirements in the software lifecycle process from the infrastructure point of view. However, within the next 2 or 3 years, they will make sure that customers will not think twice before going for cloud offerings while selecting infrastructure for application development, deployment and maintenance (Piyumal 2018). While many practical tools will become available for resolving many of the initial issues associated with serverless computing, the research potential also enjoys immense growth (Spillner 2017).

5. CONCLUSION

In this paper, a MLR method was conducted to investigate serverless computing phenomenon. To answer the research questions we retrieved different primary studies from both, grey and scientific literature. Authors found that serverless computing has the potential to revolutionize the way we deploy code. Although some concerns are appearing in serverless scenarios, organizations are moving their applications to the cloud for a wide variety of very good reasons including financial and operational ones. However, authors believe that serverless computing, specially combined with microservices architectures are the future of cloud applications and are here to stay, but also to evolve improving aspects like supporting services, availability of tools, transparency in vendor side and increasing portability among vendors.
As future works, authors propose the study of serverless computing education in higher education institutions and from the more technical side, the study of aspects like portability costs, the benchmark of maintenance aspects and the study of automated testing environments in serverless settings.

REFERENCES

Adzic, G., & Chatley, R. (2017). Serverless Computing: Economic and Architectural Impact. In Proceedings of the 2017 11th Joint Meeting on Foundations of Software Engineering (pp. 884–889). New York, NY, USA: ACM. https://doi.org/10.1145/3106237.3117767

Akiwatkar, R. (2017). The Components of a Serverless Architecture Framework - DZone Cloud. Retrieved April 2, 2018, from https://dzone.com/articles/the-components-of-a-serverless-architecture-framework

Amazon. (2018). Amazon API Gateway [Cloud vendor]. Retrieved April 2, 2018, from https://aws.amazon.com/api-gateway/

Ast, M., & Gaedke, M. (2017). Self-contained Web Components through Serverless Computing. In Proceedings of the 2Nd International Workshop on Serverless Computing (pp. 28–33). New York, NY, USA: ACM. https://doi.org/10.1145/3154847.3154849

AWS. (2018). Serverless Computing – Amazon Web Services [Cloud vendor]. Retrieved April 2, 2018, from https://aws.amazon.com/serverless/

Azure, M. (2018). Serverless Computing | Microsoft Azure [Cloud vendor]. Retrieved April 2, 2018, from https://azure.microsoft.com/en-us/overview/serverless-computing/

Baker, M., & Canonical. (2017, December 8). 2018 cloud trend predictions. Retrieved April 1, 2018, from http://www.datacenterdynamics.com/content-tracks/colo-cloud/2018-cloud-trend-predictions/99450.fullarticle

Baldini, I., Castro, P., Chang, K., Cheng, P., Fink, S., Ishakian, V., … Suter, P. (2017a). Serverless Computing: Current Trends and Open Problems. In Research Advances in Cloud Computing (pp. 1–20). Springer, Singapore. https://doi.org/10.1007/978-981-10-5026-8_1

Baldini, I., Castro, P., Cheng, P., Fink, S., Ishakian, V., Mitchell, N., … Suter, P. (2016). Cloud-Native, Event-Based Programming for Mobile Applications. In 2016 IEEE/ACM International Conference on Mobile Software Engineering and Systems (MOBILESoft) (pp. 287–288). Austin, TX, USA: IEEE. https://doi.org/10.1109/MobileSoft.2016.063

Barga, R. (2018). Serverless Computing Redefining the cloud. Presented at the Amazon Web Services. Retrieved from http://www.serverlesscomputing.org/wosc17/presentations/barga-keynote-serverless.pdf

Barik, R., Chakrabarti, A., & Pal, R. (2018, January). Serverless Computing- Architectural Considerations & Principles. Deloitte. Retrieved from https://www2.deloitte.com/content/dam/Deloitte/tr/Documents/technology-media-telecommunications/Serverless%20Computing.pdf

Boulton, C. (2018). 6 trends shaping IT cloud strategies today | CIO. Retrieved April 1, 2018, from https://www.cio.com/article/3137946/cloud-computing/6-trends-that-will-shape-cloud-computing-in-2017.html#tk.cio_rs

Calderón, A., Ruiz, M., & O'Connor, R. V. (2018). A multivocal literature review on serious games for software process standards education. Computer Standards & Interfaces, 57, 36–48. https://doi.org/10.1016/j.csi.2017.11.003

Carey, S. (2017). Predicting the Biggest Cloud Trends in 2018. Retrieved April 2, 2018, from https://www.computerworlduk.com/cloud-computing/predicting-biggest-cloud-trends-in-2018-3669252/

Castro, P., Ishakian, V., Muthusamy, V., & Slominski, A. (2017). Serverless Programming (Function as a Service) (pp. 2658–2659). Presented at the 2017 IEEE 37th International Conference on Distributed Computing Systems, Atlanta, GA, USA: IEEE. https://doi.org/10.1109/ICDCS.2017.305

Chan, M. (2017, March 13). Serverless Architectures: Everything You Need to Know. Retrieved April 2, 2018, from https://www.thorntech.com/2017/03/serverless-architectures-everything-need-know/

Chang, K. S. P., & Fink, S. J. (2017). Visualizing serverless cloud application logs for program understanding. In 2017 IEEE Symposium on Visual Languages and Human-Centric Computing (VL/HCC) (pp. 261–265). Raleigh, NC, USA: IEEE. https://doi.org/10.1109/VLHCC.2017.8103476

Colomo-Palacios, R., Fernandes, E., Soto-Acosta, P., & Larrucea, X. (2018). A case analysis of enabling continuous software deployment through knowledge management. International Journal of Information Management. https://doi.org/10.1016/j.ijinfomgt.2017.11.005

Contino. (2017). The Future of the Serverless Market [text/html]. Retrieved April 2, 2018, from https://www.contino.io/insights/the-future-of-the-serverless-market

Contino. (2018). Introduction to Serverless Computing with AWS Lambda. contino.io. Retrieved from https://www.contino.io/files/Contino-Introduction-to-Serverless-Computing-with-AWS-Lambda.pdf

Crane, M., & Lin, J. (2017). An Exploration of Serverless Architectures for Information Retrieval. In Proceedings of the ACM SIGIR International Conference on Theory of Information Retrieval (pp. 241–244). New York, NY, USA: ACM. https://doi.org/10.1145/3121050.3121086

Dwyer, I. (2016). SERVERLESS COMPUTING DEVELOPER EMPOWERMENT REACHES NEW HEIGHTS. iron.io. Retrieved from https://www.iron.io/docs/Whitepaper_Serverless_Final_V2.pdf

Eivy, A. (2017). Be Wary of the Economics of "Serverless" Cloud Computing. IEEE Cloud Computing, 4(2), 6–12. https://doi.org/10.1109/MCC.2017.32

Elmore, R. F. (1991). Comment on "Towards Rigor in Reviews of Multivocal Literatures: Applying the Exploratory Case Study Method." Review of Educational Research, 61(3), 293–297. https://doi.org/10.3102/00346543061003293

Garousi, V., Felderer, M., & Hacaloğlu, T. (2017). Software test maturity assessment and test process improvement: A multivocal literature review. Information and Software Technology, 85, 16–42. https://doi.org/10.1016/j.infsof.2017.01.001

Garousi, V., Felderer, M., & Mäntylä, M. V. (2016). The Need for Multivocal Literature Reviews in Software Engineering: Complementing Systematic Literature Reviews with Grey Literature. In Proceedings of the 20th International Conference on Evaluation and Assessment in Software Engineering (p. 26:1–26:6). New York, NY, USA: ACM. https://doi.org/10.1145/2915970.2916008

Garousi, V., & Mäntylä, M. V. (2016). When and what to automate in software testing? A multi-vocal literature review. Information and Software Technology, 76, 92–117. https://doi.org/10.1016/j.infsof.2016.04.015

Glikson, A., Nastic, S., & Dustdar, S. (2017). Deviceless Edge Computing: Extending Serverless Computing to the Edge of the Network. In Proceedings of the 10th ACM International Systems and Storage Conference (p. 28:1–28:1). New York, NY, USA: ACM. https://doi.org/10.1145/3078468.3078497

Golden, B. (2018). Serverless computing: 5 things to know about the post-container world. Retrieved April 2, 2018, from https://techbeacon.com/aws-lambda-serverless-apps-5-things-you-need-know-about-serverless-computing

Goncharov, L. (2017). Serverless architecture pros and cons | AgileEngine. Retrieved April 2, 2018, from https://agileengine.com/why-should-consider-a-serverless-architecture-for-your-next-project/

Google. (2018a). Datastore - NoSQL Schemaless Database | Google Cloud [Cloud vendor]. Retrieved April 2, 2018, from https://cloud.google.com/datastore/

Google (2018b). Serverless Architecture | Google Cloud [Cloud vendor]. Retrieved April 2, 2018, from https://cloud.google.com/serverless/

IBM, (2018). Serverless web application and API [Cloud vendor]. Retrieved April 2, 2018, from https://console.bluemix.net/docs/tutorials/serverless-api-webapp.html#serverless-web-application-and-api

Kanso, A., & Youssef, A. (2017). Serverless: Beyond the Cloud. In Proceedings of the 2Nd International Workshop on Serverless Computing (pp. 6–10). New York, NY, USA: ACM. https://doi.org/10.1145/3154847.3154854

Kiriaty, Y. (2016). Announcing general availability of Azure Functions. Retrieved March 6, 2018, from https://azure.microsoft.com/en-us/blog/announcing-general-availability-of-azure-functions/

Leitersdorf, Y., Ventures, Y., Schreiber, O., Reznikov, I., & Ninyo, I. (2017). The big opportunities in serverless computing | VentureBeat. Retrieved April 2, 2018, from https://venturebeat.com/2017/10/22/the-big-opportunities-in-serverless-computing/

Lloyd, W., Ramesh, S., Chinthalapati, S., Pallickara, S., & Ly, L. (2017). Serverless Computing: An Investigation of Factors Influencing Microservice Performance . Retrieved March 27, 2018, from https://www.semanticscholar.org/paper/Serverless-Computing%3A-An-Investigation-of-Factors-Lloyd-Ramesh/6c778f7dfcde0e2508fd9c35a6ebb5b4ff58107e

Lynn, T., Rosati, P., Lejeune, A., & Emeakaroha, V. (2017). A Preliminary Review of Enterprise Serverless Cloud Computing (Function-as-a-Service) Platforms. Presented at the Cloud Computing Technology and Science (CloudCom), 2017 IEEE International Conference on, Hong Kong, China: IEEE. https://doi.org/10.1109/CloudCom.2017.15

Malawski, M., Gajek, A., Zima, A., Balis, B., & Figiela, K. (2017). Serverless execution of scientific workflows: Experiments with HyperFlow, AWS Lambda and Google Cloud Functions. In Future Generation Computer Systems. Elsevier. https://doi.org/10.1016/j.future.2017.10.029

Maruti, T. (2017, June 5). Serverless Architecture The Future of Business Computing. Retrieved April 2, 2018, from https://www.marutitech.com/serverless-architecture-business-computing/

Mcdonald, E. (2017, November 3). Why Serverless Architecture Is Called as A Game Changer? Retrieved March 19, 2018, from http://evavmcdonald8.blogspot.com/2017/11/why-serverless-architecture-is-called-as-a-game-changer.html

McGrath, G., & Brenner, P. R. (2017). Serverless Computing: Design, Implementation, and Performance. In 2017 IEEE 37th International Conference on Distributed Computing Systems Workshops (ICDCSW) (pp. 405–410). Atlanta, GA, USA: IEEE. https://doi.org/10.1109/ICDCSW.2017.36

Myrbakken, H., & Colomo-Palacios, R. (2017). DevSecOps: A Multivocal Literature Review. In Software Process Improvement and Capability Determination (pp. 17–29). Springer, Cham. https://doi.org/10.1007/978-3-319-67383-7_2

Ogawa, R. T., & Malen, B. (1991). Towards Rigor in Reviews of Multivocal Literatures: Applying the Exploratory Case Study Method. Review of Educational Research, 61(3), 265–286. https://doi.org/10.3102/00346543061003265

Pérez, A., Moltó, G., Caballer, M., & Calatrava, A. (2018). Serverless computing for container-based architectures. Future Generation Computer Systems, 83, 50–59. https://doi.org/10.1016/j.future.2018.01.022

Persson, P., & Angelsmark, O. (2017). Kappa: Serverless IoT Deployment. In Proceedings of the 2Nd International Workshop on Serverless Computing (pp. 16–21). New York, NY, USA: ACM. https://doi.org/10.1145/3154847.3154853

Piyumal, M. (2018, January 23). Will serverless be the future of enterprise application development? Retrieved April 2, 2018, from https://medium.com/@manjulapiyumal/will-serverless-be-the-future-on-enterprice-application-development-9a1901798b0a

Rai, G., Pasricha, P., Malhotra, R., & Pandey, S. (2017, May 24). Serverless Architecture: Evolution of a New Paradigm. Retrieved April 1, 2018, from https://www.globallogic.com/gl_news/serverless-architecture-evolution-of-a-new-paradigm/

Rapyder. (2018, February 13). 15 Cloud Computing Predictions to Watch in 2018. Retrieved April 2, 2018, from http://www.rapyder.com/15-cloud-computing-predictions-watch-2018/

Roberts, M. (2016, August 4). Serverless Architectures [article]. Retrieved March 5, 2018, from https://martinfowler.com/articles/serverless.html

Sampé, J., Sánchez-Artigas, M., García-López, P., & París, G. (2017). Data-driven Serverless Functions for Object Storage. In Proceedings of the 18th ACM/IFIP/USENIX Middleware Conference (pp. 121–133). New York, NY, USA: ACM. https://doi.org/10.1145/3135974.3135980

Savage, N. (2018). Going Serverless. Commun. ACM, 61(2), 15–16. https://doi.org/10.1145/3171583

Sbarski, P. (2017a). Serverless Architectures on AWS: With Examples Using AWS Lambda. Manning.

Sbarski, P. (2017b). The essential guide to serverless technologies and architectures. Retrieved April 1, 2018, from https://techbeacon.com/essential-guide-serverless-technologies-architectures

Serverless. (2018). Why Serverless? Retrieved April 1, 2018, from https://serverless.com/learn/

Spillner, J. (2017). Practical Tooling for Serverless Computing. In Proceedings of the10th International Conference on Utility and Cloud Computing (pp. 185–186). New York, NY, USA: ACM. https://doi.org/10.1145/3147213.3149452

Stamat, A. D. (2016, July 7). What is Serverless Computing and Why is it Important [Cloud vendor]. Retrieved April 2, 2018, from https://blog.iron.io/what-is-serverless-computing/

Tata. (2017). Serverless Computing: A Compelling Opportunity for Today’s Digital Enterprise. Tata Consultancy Services Limited. Retrieved from https://www.tcs.com/content/dam/tcs/pdf/technologies/cloud/abstract/Serverless%20Computing.pdf

van Eyk, E., Iosup, A., Seif, S., & Thömmes, M. (2017). The SPEC Cloud Group’s Research Vision on FaaS and Serverless Architectures. In Proceedings of the 2Nd International Workshop on Serverless Computing (pp. 1–4). New York, NY, USA: ACM. https://doi.org/10.1145/3154847.3154848

Varghese, B., & Buyya, R. (2018). Next generation cloud computing: New trends and research directions. Future Generation Computer Systems, 79, 849–861. https://doi.org/10.1016/j.future.2017.09.020

Watts, S. (2018). What is Serverless Computing? Serverless Computing Explained – BMC Blogs. Retrieved April 2, 2018, from https://www.bmc.com/blogs/serverless-computing/

Wolf, O. (2018). What is the relationship between Serverless Computing and microservices? - Quora. Retrieved from https://www.quora.com/What-is-the-relationship-between-Serverless-Computing-and-microservices#VEvPi

Yan, M., Castro, P., Cheng, P., & Ishakian, V. (2016). Building a Chatbot with Serverless Computing. In Proceedings of the 1st International Workshop on Mashups of Things and APIs (p. 5:1–5:4). New York, NY, USA: ACM. https://doi.org/10.1145/3007203.3007217

Yang, K. (2017, October 23). Tech Primer: Serverless Computing Converging FaaS. Retrieved April 12, 2018, from http://www.dataversity.net/tech-primer-serverless-computing-converging-faas/