computers

MDPI

Review

Smart Healthcare System in Server-Less Environment: Concepts, Architecture, Challenges, Future Directions

Rup Kumar Deka [1], Akash Ghosh [2], Sandeep Nanda [2], Rabindra Kumar Barik [2] and Manob Jyoti Saikia [3,*]

[1] School of Computer Science and Engineering, Vellore Institute of Technology, Vellore 632014, Tamil Nadu, India
[2] School of Computer Applications, KIIT Deemed to be University, Bhubaneswar 751024, Odisha, India
[3] Department of Electrical Engineering, University of North Florida, Jacksonville, FL 32224, USA
* Correspondence: manob.saikia@unf.edu

Abstract: Server-less computing is a novel cloud-based paradigm that is gaining popularity today for running widely distributed applications. When it comes to server-less computing, features are available via subscription. Server-less computing is advantageous to developers since it lets them install and run programs without worrying about the underlying architecture. A common choice for code deployment these days, server-less design is preferred because of its independence, affordability, and simplicity. The healthcare industry is one excellent setting in which server-less computing can shine. In the existing literature, we can see that fewer studies have been put forward or explored in the area of server-less computing with respect to smart healthcare systems. A cloud infrastructure can help deliver services to both users and healthcare providers. The main aim of our research is to cover various topics on the implementation of server-less computing in the current healthcare sector. We have carried out our studies, which are adopted in the healthcare domain and reported on an in-depth analysis in this article. We have listed various issues and challenges, and various recommendations to adopt server-less computing in the healthcare sector.

Keywords: server-less computing; FaaS; smart-health; healthcare systems

Citation: Deka, R.K.; Ghosh, A.; Nanda, S.; Barik, R.K.; Saikia, M.J. Smart Healthcare System in Server-Less Environment: Concepts, Architecture, Challenges, Future Directions. *Computers* **2024**, *13*, 105. https://doi.org/10.3390/computers13040105

Academic Editor: Wenbing Zhao

Received: 26 March 2024
Revised: 12 April 2024
Accepted: 15 April 2024
Published: 19 April 2024

1. Introduction

Cloud computing has emerged as a very successful technology in the era of digitized services on the Internet in the past ten years. A handful of popular emerging public cloud computing providers includes Amazon Web Services (AWS) [1], Google Cloud [2], Microsoft Azure [3], IBM Cloud [4], etc. Some of the cloud management enterprises that allow the administrators to create state-of-the-art cloud infrastructure are Open-Stack [5], Open Nebula [6], Server-space, etc. Although it is well-liked and the foundation of many applications, it is linked to many kinds of problems and confronts a wide range of difficulties. A specific cloud solution that was just recently created is called server-less computing [7,8]. With the help of cloud computing, all configuration, provisioning, and management tasks are now the duty of the server. The cloud technology built around virtual machines and containers is the server-less computing architecture. Hardware is visualized on an operating system that hosts it via server-less computing, which is based on virtual machines. Server-less computation with containers requires the construction of numerous distinct environments to manage various workloads concurrently on the operating system that serves as the host. In the past few years, container-based server-less technology has gained a lot of popularity [9]. Over the conventional, antiquated cloud computing approaches, server-less computation has several advantages. For a typical individual, it is far more difficult to effectively apply the intricate nature of the more modern structures [3]. AWS Lambda is a technology made available by Amazon Web Services that enables customers to create, deploy, and make use of server-less platforms for their applications. Developers must upload a collection of happenings that are tied to a stateless function and may be used as

triggers. Examples of events include making HTTP requests to preset endpoints configured by the AWS Application Programming Interface (API) Gateway service or transferring a file to a bucket in Amazon S3 (Simple Storage Service). Function-as-a-service (FaaS) platforms are utilized in server-less computing to handle all operational as well as computational tasks, including resource management, scalability, function deployment, and monitoring. Engineers are given the flexibility to concentrate solely on the company's business logic operations without becoming involved in the creation of applications. Using the server-less computing approach, programmers may execute event-driven code autonomously without maintaining or configuring servers, changing the design and development of contemporary scalable systems.

1.1. Smart Healthcare Market

The smart healthcare market is experiencing a rise in leveraging new technologies, improving the efficiency, quality, and accessibility of healthcare services. The increasing adoption of digital health solutions, the growing prevalence of chronic diseases, and the need for remote monitoring and telemedicine services are the main reasons for these developments. Some of the key trends and technologies driving the smart healthcare market include the following [10–14]. In Figure 1, we can see the future of the smart healthcare market (https://www.linkedin.com/pulse/smart-healthcare-market-towards-healthcare-1f, accessed on 12 February 2024).

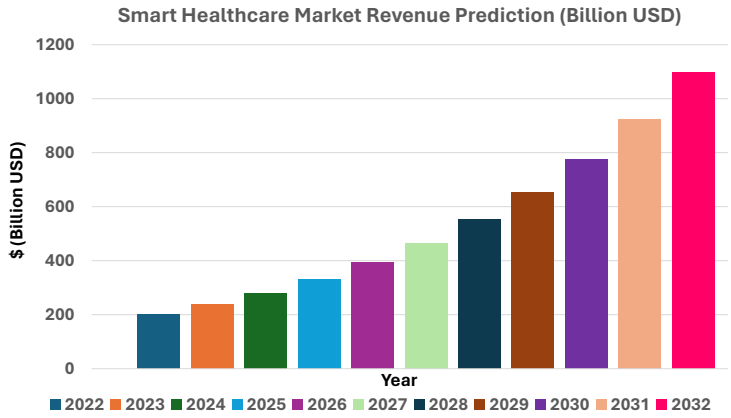

Figure 1. Statistical observation of smart healthcare market: based on a study published by Towards Healthcare, the global smart healthcare market size is expected to grow from $201.83 billion in 2022 and is estimated to be worth $1097.27 billion by 2032, growing at a CAGR of 18.5% between 2023 and 2032.

- Remote Healthcare allows healthcare providers to deliver care remotely, reducing the need for in-person visits and improving access to healthcare services, especially in rural or remote areas. Remote monitoring technologies, such as wearable and connected devices, enable continuous monitoring of patients' health parameters and facilitate early intervention.
- The use of Artificial Intelligence (AI) and Machine Learning (ML) to analyze large datasets and derive insights can improve diagnostics, personalize treatment plans, and enhance operational efficiency in healthcare settings.
- The Application of the Internet of Medical Things (IoMT) can be used to collect and transmit healthcare data. IoMT devices include wearable, implantable devices, and home monitoring devices, which help in real-time health monitoring and management.
- Enhanced accountability can be attained by using blockchain technology, which is being explored to secure health data exchange. It can also help in tracking the authenticity and integrity of medical records.

- Cloud computing enables healthcare providers to store, manage, and access large volumes of data securely and cost-effectively. It also facilitates collaboration among healthcare professionals and supports the deployment of scalable healthcare applications.
- Smart hospitals integrate advanced technologies to enhance patient care, streamline operations, and improve efficiency by resolving a few specific issues like staff scheduling [15], patient admission scheduling [16], home healthcare routing [17], and nurse rostering [18].

1.2. Function as a Service (FaaS) Computing Model

Each unique job within a server-less system can be represented via a standalone function that acts as its processing unit. Every operation can take the shape of a package and execute in a separate environment. Resource allocation and provisioning for particular functions instead of environments have become the norm for providers of services. Therefore, a server-less computing application may be thought of to be an assembly of separate, stateless services. By triggering a few events using a Web API, an individual can access the service. Users do not have to be concerned about the execution environment and resource settings. Similarly to entities in object-oriented programming frameworks and functions based on practical programming principles, methods in server-less systems. The cloud-based FaaS (Figure 2) service paradigm is crucial to the online delivery of healthcare-related operations. Since each health service is packaged in a stateless method, it is portable and may be used in any location. Events often trigger the execution of functions like a service. The event information and company logic must be combined before being uploaded to the internet by the developer. The product or service provider has charge of the remaining administration, setup, and operational environment [19]. To do function-based programming in various computing environments, we can find different abilities and issues as mentioned in Table 1.

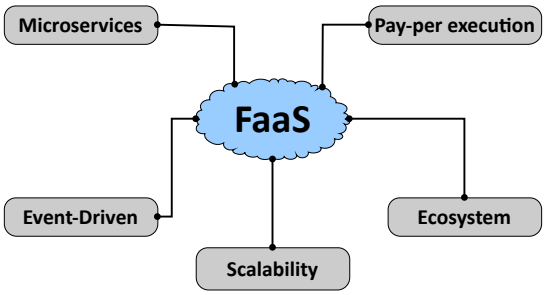

Figure 2. Ecosystem of Function-as-a-Service (FaaS).

Table 1. Function Based Programming Model.

Programming Models for Functions		
Cloud-based FaaS	Edge-based FaaS	Fog-based FaaS
Execution is centralized	Execution in edge node	Execution in cloud and edge
Input and output bind in every event	Input and output bind in per topic	Input and output bind in every selected entity
Granularity of the task is none	Granularity of the task is none	Granularity of the task is definable
Migration is possible	Migration not possible	Migration is possible
Service level objective is none	Service level objective is none	Service level objective definable
Per event, the trigger happens	Per edge trigger happens	based on the availability of selected entities trigger happens

1.3. Motivations

Depending on the demands made by the customer, server-less providers offer a variety of services, including on-demand, on-reserve, and instance services. On-demand services have become the foundation of the healthcare sector since they are significantly more trustworthy and capable of meeting demand at their highest at any given moment. Another factor to keep in mind is that different applications in the field of health care, which include electrocardiogram (ECG), electroencephalogram (EEG), E-Blood Analysis, etc., have distinct price models.

- Healthcare management systems are becoming increasingly computerized as of late, necessitating the development of quick, affordable, and dependable medical service alternatives. Because of its cost structure, auto-expansion, and versatility characteristics, a server-less framework might be a fantastic option for the healthcare industry. The importance of server-less computing has been focused on and detailed.
- Server-less computing has emerged along with other new technologies, like, cloud computing, fog computing, edge computing, IoT, artificial intelligence, etc. Similarly, Applications in medicine that require growth quickly, be immediately ready, and complete high-latency activities in a matter of seconds may find server-less technology akin to a magical tool.
- A rigorous study on various existing approaches with respect to server-less computing, related to the healthcare system has been performed and properly demonstrated through a table.
- A lot of real-time applications use server-less architectures. Nevertheless, a deeper investigation of this unique technology's interoperability with the medical sector is required. Keeping this in mind, we have shown a framework of server-less computing and made a few recommendations to resolve the possible issues and challenges.

1.4. Contributions

In this work, we have contributed the following points.

- We provide studies of various existing server-less computing approaches related to the healthcare domain.
- Possible issues and challenges have been put forward for readers and the list can be further enumerated.
- A few recommendations along with a proposed framework for server-less computing have been portrayed.

1.5. Organizations

The organization of this article is as follows. At first, we introduce the idea of a smart healthcare system in a server-less environment in Section 1 along with motivations and contributions. In Section 2, we discussed existing works of literature on this relevant research idea. We try to describe the major concerns about the healthcare management system in the current world in Section 3. After that, we discussed the need for security and privacy in smart healthcare systems, in Section 4. In Section 5, current developments in the field of server-less computing along with new techniques and possible impact on the healthcare system have been detailed. Few comparable factors are listed in Section 6, and issues & challenges are pointed out in Section 7. Section 8 is about various points of the whole research, depicting an ideal server-less architecture and recommendations. At last Section 9 provides the conclusion of our review of smart healthcare systems in server-less computing environments.

2. Related Work

Clients and programmers may simply post and execute code using a server-less framework, a revolutionary approach to application creation, not thinking over servers or resources [20]. In the modern age of cloud computing services, this is often referred to as the Function-as-a-Service delivery model. The service supplier receives responsibilities from

the programmer for architecture setup, server administration, security repair, and updating. Just concentrate on creating a company rationale. In comparison to previous distribution models, this usage-based payment-based design has also been demonstrated to be more affordable [21]. Users are not charged for downtime. When the program is operating server-side, you just pay to access the precise resources used. The FaaS concept is the foundation of server-less design. This enables users to concentrate on developing their applications as a result of a collection of distinct features that can contain any functionality, based on the precise assignment of your nomination. Every one of these tasks requires a series of occurrences to be posted to the clouds for it to take effect. Events might be alerts from all the IoT gadgets, requests via HTTP, a lot of updates to the database, and so forth [22,23]. A server-less source executes an operation on an active server or docker when it is activated [24]. You can provide an additional server to operate when you do not already have one. Every trigger should go via a gateway to the API which manages a certain aspect related to the previously specified application. Scheduling, auto-scaling, or logging events might have been resource-related. The cloud service provider responds appropriately to the request sent by API Gateway. Which operation unit or docker to set up for handling a request is decided by a scheduler. When an application requires more resources, the automatic scaling engine kicks into action. The associated operations are performed after configuring the operating environment. Typically, the client end is separated from the working environment. All assigned resources are reallocated and the generated ecosystem is removed when the program has finished functioning. The state that exists between calls to an operator is discarded. Furthermore, it goes by the name stateless operation environment. The server-less technology architecture's procedure for execution is depicted in the schematic displayed in Figure 3 [25].

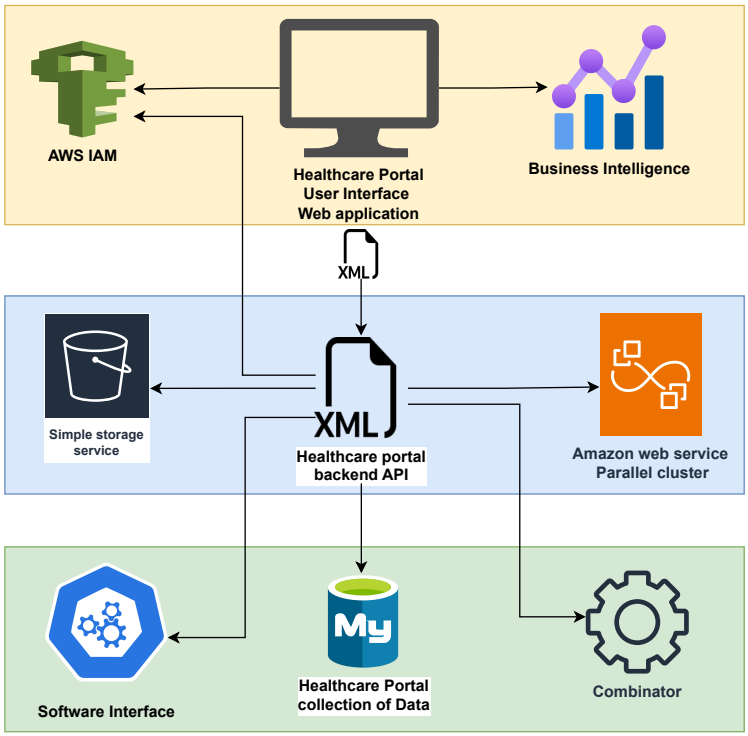

Figure 3. Server-less architecture for healthcare management system.

The AWS Cognito functionalities may be used to secure your online apps. offers both new and current clients verification and safety services. It can be altered to increase the quantity of existing users. The outer layer of your online application might have a customized user interface (UI) thanks to CloudFront. The Amazon Web Service's CloudFront element is utilized like a network for distributing content to handle and disperse material in many different ways [26]. Even though the app resides elsewhere, it enables individuals

to access material that has lower responsiveness by caching information nearer to the client. In the background, unstructured materials like analyzing images and patient data may be stored using Amazon's cloud-based storage platform S3 [27]. The server-less framework is useful for a variety of activities. The analysis of non-linear as well as linear information is crucial for the healthcare sector. Figure 3 illustrates the structure for server-less framework-powered online healthcare apps. VueJs, a JavaScript-based utility, controls the Internet-based application's front-end programming functionality [28]. Due to its ability to manage programs, it proved to be more effective compared to various tools. Both dynamic and static regions are supported by this program. NuxtJs controls the creation of stationary domains using VueJs utilities and is connected with the cloud storage service Amazon S3. At a relatively minimal cost, it provides static web pages. It could, however, vary based on how large your program is. The benefit of S3 on Amazon Web Services is the fact it allows for dynamic applications, so they go through numerous upgrades over time. It offers several interfaces that could be utilized to gather and analyze health data, and they may be put to use by executing lambda functions. The REST interface is often used to invoke server-less services [29]. The vast quantity of organized information that may be obtained by a user on the website's back-end in SQL table/XML formats should be handled by integrating Dynamo Database alongside AWS S3, similar to a server-free NOSQL database.

Lambda operations handle the Internet application's general processing. A crucial component in a server-less technology operation acts as a kind of service. Any inquiries from the API route must be utilized by the client to trigger actions. The proper function is enabled and run whenever a condition is triggered to act. Fixed file modifications are either transmitted to Amazon S3 cloud storage or the Dynamo Database where the latest graphic evaluation is saved. In this design, their Lambda method may be used to organize, search, and evaluate health data kept in the Dynamo Database, or S3 on Amazon Web Services. Data about patients can also be uploaded and stored by healthcare providers. You may use Amazon Web Services Cognito to inspect the diagrams that are produced as a consequence of the analysis that has been carried out on the system's back end. You may look at the program's analytic section for an improved visualization [19].

3. Major Concerns in Healthcare Management Systems

We frequently discover that immigrant medical providers are unclear about how to start their tough migration journey and put their products in the data center. Companies first assess the knowledge and assets required to grow their cloud-based model solutions. It is unsettling to consider adapting and disassembling a general solution within a fast-paced setting. Second, since their local virtual servers are no longer directly in charge of their cloud, enterprises may be keen on preserving and attaining cloud compliance (particularly regarding FDA CFR Part 11 and HIPAA). The third, though not final, aim is cost minimization. To overcome the aforementioned issues, health care administration systems are built on three components as follows:

3.1. Enhancing Speed of Delivery and Scalability

When implemented appropriately, a system without servers has several benefits. These are server-less programs that grow as they are used or demand rises but are not autonomously deployed. A few two-week iterations may also deploy technology and functionalities into the cloud environment. Typically, it involves the use of infrastructure, including business logic, which symbolizes the setting up and administration of the infrastructure through the use of implemented code enables conversion control, evaluation, and reversal. Amazon Web Services (AWS) CloudFormation, Terraform, Pulumi, and Microsoft Resource Manager are a few examples of such architectures. You may implement a server-less, extensible solution throughout your internet-based health ecosystem. Both the Medicare and Medicaid programs, for instance, maintain technology that can handle thousands of clients at once using cloud services provided by AWS. The same potent AWS features may be used by businesses dealing with massive datasets to improve

data collecting, safe storage, efficient processing, and information libraries for big data analytics [19].

3.2. Ensuring Compliance

Through initiatives like the partnership toward efficient server-less cloud layouts, the medical industry is boarding the technological train. The basic objective of a method like AWS, for instance, is to provide as many amenities as it can while upholding compliance with HIPAA along with additional regulations. Covered companies may use the Amazon Web Services (AWS) HIPAA scheme. The Health Insurance Portability and Accountability Act of 1996 (HIPAA), in addition to collaborators in business, handles, controls, and safeguards secured medical data using the safe atmosphere of AWS. It plans to bring additional services, including server-less technologies, into the HIPAA framework in response to customer input [19].

3.3. Cost Optimization

Customers can quickly download and upgrade items because there is no technology to set up or administer, which lowers costs and lengthens product life. The complexity of equipment is frequently quite great, and the unreasonably high and stagnating monthly expenses for repairs. Agile Vicert is dedicated to improving their client's digital medical solutions, and migrating onto the cloud is going to boost speed and endurance while also sharply lowering the expenses of care. A client's annual availability for service has risen to 99.9% despite significant application outages. Amazon Web Services Lambda, Microsoft Azure Functions, Open Whisk, and other server-less technologies designed specifically for cloud services offer both better safety and quicker updates, leading to reduced expenses, simpler and faster deployments, and increased dependability, and lifespan of service [19].

3.4. Scheduling

The program or additional function of the consumer sends a call query to the supplier. Frequently, these petitions have due dates. For instantaneous, delay-sensitive, or security-critical tasks, this is particularly crucial. Vendors must prepare ahead for the function's timing and location (i.e., the computational node within which it will operate), as well as other system-wide factors like resource use and energy consumption. Vendors employ a variety of planning techniques while executing features. The following is a summary of these tactics, a classification of planning strategies for server-less technology. Keep in mind that the organizer may employ many of these, simultaneously, in practice.

4. Security and Privacy

Security is a key concern for any computing service, server-less or not. server-less and other cloud services face various security challenges [30]. Here we consider security issues that specifically threaten the normal operation of server-less services. User privacy is also a consideration in such an environment. Figure 4 shows a taxonomy of security approaches for server-less computing.

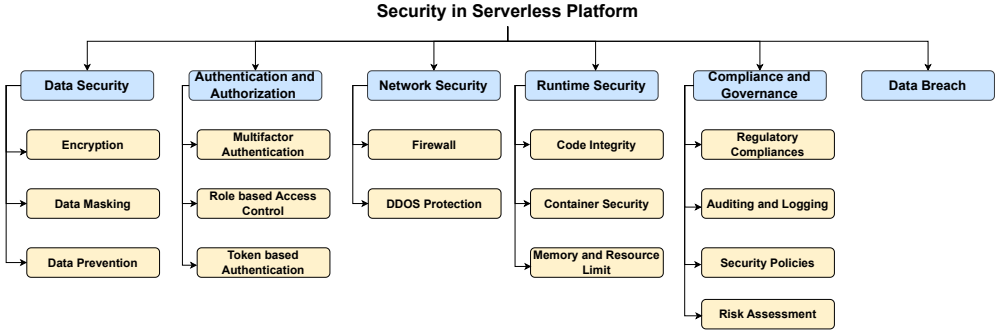

Figure 4. A taxonomy of security in Server-less platforms.

4.1. Authentication and Authorization

Authenticating apps to ensure that only valid programs may utilize accessible functionalities is the main security concern in server-less computing systems. Free riders might exploit the victim's assets without identification. Utilizing tokens for authentication in response parameters is a typical strategy for preventing these attacks.

Such a system is now implemented by Amazon's Lambda, which establishes the identity of the caller via bearer tokens' authentication mechanism. The token might be readily extracted and used in another program, though, if the inquiry is made over an unsecured channel. The SSL or TLS encryption protocol may be used to counteract this kind of danger. However, program's hardware may be unable to support these complex public key-based methods. Extremely low-power Internet of Things gadget is an illustration of such devices. For these situations, request marking with less resource use has been previously suggested [31]. However, we think that developing and executing safety protocols with low power needs for the Internet of Things and embedded gadget authentication is an intriguing area for further study. Replay attacks can also be used against server-less services. In this kind of assault, the perpetrator is more concerned with the results of delivering the messages than with their substance. As a result, hackers can replay demands to perform safe procedures after intercepting them to obstruct regular system operation. This kind of assault can involve repeatedly requesting users to log out keeping them from using the service they have asked to access. Implementing a prevention and identification strategy against this kind of assault is necessary. There are additional challenges with permission, which involve limiting access for other individuals and designating exactly who or functions can call particular operations. This is distinct from the previously described application-level security. Here, we permit somebody or an algorithm to execute a different function. Application security may be seriously jeopardized by improper authentication methods. Keep in mind that the availability of standard functionality constitutes one of several anticipated advantages of the server-less platform described previously. Functionality is accessible outside the application proprietor's permission in the absence of a system for authentication. Access control using roles is the method that Amazon Lambda utilizes to handle this significant problem. Customers dynamically define roles having a certain number of permissions capabilities. This method has the disadvantage of only supporting one feature and not supporting authorization at the workflow stage [32].

4.2. Runtime Security

Application susceptibility to typical settings for execution has emerged as one of the main security issues in the wake of both Specter [33] and Meltdown [34] vulnerabilities. Because multiple functions with numerous owners are executed in a communal setting, this issue is particularly severe in server-less setups. We offer a quick and effective JavaScript engine. It employs an encrypted enclave for its operating environment to thwart this kind of assault [35]. Only supporting JavaScript and having a large memory footprint constitute the work's constraints. The scientific community should pay special attention to this field of study.

The operating environment may additionally be modified to keep an eye on processes in use and to spot and stop harmful activities. Runsec, an altered docker runtime, is introduced by SecLambda [36]. It records HTTP requests as well as input and output operations in an attempt to figure out if services comply with a predetermined set of confidentiality regulations. Trapeze [37] uses a similar approach, which is to isolate each server-less operation within an isolated environment and to block all communications between the service and the outside world by the guidelines for enforcing policies set forth by an evolving data-restricted model. Valve [38] suggests a server-less architecture that imposes constraints around the absence of data and real-time tracking. These methods have the drawback of having a substantial efficiency effect on server-less systems. The solution is to adopt a lighter solution for delay-sensitive functions.

4.3. Resource Exhaustion Attacks

The main goal of this assault is to interrupt service or place excessive economic or financial stress on the person in question by abusing their resources. Clients and suppliers may both be targets of such assaults. A hacker might use the program to send the supplier erroneous requests. Although automatic scaling server-less services could cope with these circumstances, the supplier may deny more requests or perhaps at the very least place a substantial cost on the program owner if demand exceeds their contract with them. Establishing a monitoring strategy for your server-less company is necessary to identify and prevent this kind of assault. Attacks based on resource depletion are also possible against the vendor. For mid-sized and small vendors in particular, this kind of assault would be disastrous. By observing how the system behaves, a hacker in this situation has the option of learning about or abusing the supplier's internal workings. An attacker may begin a chain of strikes by purposefully obstructing optimization attempts if they are aware of this and choose to do so. For instance, if a hacker is aware of a vendor's packing strategy, they might introduce fabricated dependency into additional sources of data to avoid functions from spreading near the information source, thereby having a significant negative impact on the vendor's networking.

4.4. Privacy Issues

There are numerous uses for cloud-based services that deal with data protection, particularly in the Internet of Things (IoT) arena. User confidentiality, for instance, is crucial for medical apps that gather patient information and utilize it to make judgments. In contrast to security assaults, the objective of a hacker is not to alter the system's typical behavior. Instead, it makes an effort to infer insights regarding a person or several users utilizing the bare minimum of data that has been gathered. An attacker could be keen to learn whether or not a user is healthy, for instance, in the Internet of Things (IoT) healthcare sector. An assailant can acquire plenty of context-related data to conclude an innocent person. If the Internet merely makes use of application-level privacy protocols, they are particularly doable. These are a few instances of surrounding information that might expose private data concerning a victim:

- Whatever method is invoked. For instance, if the gate-accessible function is called within a server-less surveillance system, a hacker might assume an intrusion.
- Many function calls. For instance, a set of features in an electronic health record may offer details on how well a person is doing. The method is invoked when and to where. For instance, an increasingly common setting that is appealing to the business's rivals may be revealed by a web-based shop.
- Speed of function calls. The monitoring system mentioned before may disclose private information regarding a disaster at the entrance.

Comprehensive confidentiality and disguise methods must be used for server-less services to avoid privacy violations.

5. Current Developments in Server-Less Computing

We can see lots of development with respect to server-less computing, as tabulated in Table 2.

Table 2. Current Developments in the Relevant Server-less Architecture Field.

Authors' Name	Year	Purpose	Technology Involved	Application Domain	Conclusive Remark	Reference
Nastic et al.	2017	Real-time analysis of healthcare data using server-less computing environment, edge computing, and cloud support.	Server-less Computing, Edge Computing, Cloud Supported	Data Analytic, Healthcare	Proposed server-less data analysis model can provide better results overcoming the cloud and edge computing. But still, it is very challenging to overcome the elasticity of the cloud platform.	[39]
Chinchole et al.	2017	Designing a cloud-based messaging application to deliver medication-related information in rural areas	Cloud Computing	Healthcare	The proposed application enables users to understand and obain information regarding the required medication and order it online.	[40]
Iyengar et al.	2018	preserving privacy which is essential for healthcare applications that deal with confidential data	Cloud Platforms	Healthcare	Security and Privacy is one of the main concerns for the client specifically in the case of the healthcare domain.	[19]
Al-Masri et al.	2018	IoT-based urban waste management using server-less architecture	IoT, Server-less Computing	Waste Management	Detection of waste disposal violation in real time using IoT-based edge computing framework.	[41]
Ergüzen and Mahmut	2018	Designing a model to store medical images using distributed file system structure to provide robust, available, scalable, and server-less solution structure.	IoT, Big Data, Distributed File Systems, Server-less Solutions	Medical Field	Survival of the system can be ensured, and security is provided using the proposed model.	[42]
Niu et al.	2019	Experimental analysis of protein sequence comparison using server-less computing, i.e., Amazon Lambda in the cloud platform.	Cloud Computing, server-less Computing	Biomedical research	Protein Sequence alignment, analysis, and comparison are better in server-less computing environments leveraging 100s of CPUs and computational power. And it is proven to be better than GPUs.	[43]
Crespo-Cepeda et al.	2019	Experimental analysis of understanding high-throughput applications of Bio-informatics in a server-less computing environment and understanding the usage of resource management.	Servers-less Computing	Biomedical research	Using cloud technology one can face problems in storage handling, processing of the data, proper integration of information, and understanding omics and clinical data. These problems can resolved using a server-less computing paradigm.	[44]
Pérez et al.	2019	Designing programming model and middleware to understand high throughput application in server-less computing	server-less Computing	Applied Computing, Medical Image Analysis	server-less computing can be the optimal choice for cost-effective execution of loosely coupled tasks provided by AWS Lambda.	[45]
Marefat and Juneja	2019	Patient-specific Arrhythmia Detection in server-less paradigm	Data parallelization, Deep Learning, server-less paradigm	Healthcare	Data parallelization in server-less paradigm increases the execution speed up, which helps the deep learning architecture	[46]

Table 2. *Cont.*

Authors' Name	Year	Purpose	Technology Involved	Application Domain	Conclusive Remark	Reference
Paul et al.	2019	Designing real healthcare monitoring architecture in server-less paradigm.	server-less Architecture, Cloud Computing	Healthcare	server-less computing also helps the developer to build a large application using Function as a Service without thinking about the management and scalability of the infrastructure	[47]
Cheng et al.	2019	Understanding FaaS in fog computing environment along with server-less architecture.	Fog Computing, server-less Computing, IoT	Data Computation	Combining ideas of server-less and fog computing, system efficiency can improve and service latency can be reduced	[48]
kaffes et al.	2019	Studying the necessity of cluster-based centralized granular scheduling for server-less functions	server-less Computing	Scheduling Algorithm	Cluster-level scheduler for server-less functions to enhance the elasticity and reduce interference.	[49]
Eapen et al.	2020	To implement a digital healthcare system using machine learning, demonstrated a four-tier architecture to support scalability, portability, and discoverability	Machine Learning, Cloud Computing	Healthcare	Patient-centric medical health solutions to enable the client to make properly informed decisions, and thus ML and AI are becoming increasingly prevalent.	[50]
Pandey et al.	2020	Detection and Notification of health-related information like heartbeat and blood pressure, in the mobile device.	Cloud Computing	Healthcare	Proposed detection system encapsulates security, privacy, protection, and efficiency.	[51]
Trilles et al.	2020	Understanding the need for an IoT platform based on micro-services and server-less paradigm to support smart farming	IoT, server-less Computing	Smart Farming	Proposed Architecture can handle and discover heterogeneous IoT devices, management of data and event, scalability, re-usability, interoperability, reliability, availability, and security.	[52]
Grzesik and Mrozek	2021	Understand the applicability of base-calling nanopore read in server-less computing environment for multiple sequencing.	server-less Computing	Bio-informatics	In the field of bio-informatics, Amazon Lambda server-less computing provides a proper environment to apply base-calling nanopore reading from multiple sequences while maintaining low infrastructure overhead.	[53]
Benedetti et al.	2021	A performance study in terms of resource consumption and latency is presented for the warm and cold-start deployment mode, and implemented using Open FaaS	server-less Computing, IoT	Performance Evaluation	Cold start and warm start, are two specific issues I server-less computing environment and need to be dealt it with as per user demand.	[54]

Table 2. *Cont.*

Authors' Name	Year	Purpose	Technology Involved	Application Domain	Conclusive Remark	Reference
he et al.	2022	Building annotated corpus using server-less annotator tool, i.e., Mediator.	server-less Computing	Bio-informatics	Without installing any runtime environment, using MedTator, a server annotation tool, can annotate rapid corpus development	[55]
Grzesik et al.	2022	Understanding the difference of usability between cloud computing and server-less computing for integrative analysis of multiple omic data sources.	Cloud computing, server-less Computing	Bio-informatics	server-less computation becoming an increasingly popular choice for bio-informatics research. It can provide decreased processing time, cost-optimization, less maintenance overhead, better parallelization, and reliable privacy of processed data.	[56]
Sadek et al.	2022	Design and implementation of the medical searching system in server-less paradigm	server-less Computing	Medical Field	Micro-services and server-less paradigm are emerging and providing better results and also in this case, creating a medical data searching application, i.e., scanMedicine, providing help to health care professionals.	[57]

5.1. Server-Less Computing and IoT

The Internet of Things (IoT) is a widely used technology to solve various context-specific problems in the modern era. All these services can be provided through an organized distributed platform using IoT. In general, the main phases of IoT are collecting data using sensors and monitoring and analyzing the data using predefined functions or modules. Collected data can be injected through various applications as per different required services.

- Scalability is one of the inherent properties of server-less computing, which can easily handle the varying workloads of IoT devices. Functions can be triggered based on IoT events, such as sensor data updates, without the need to provision or manage servers.
- Random and unpredictable workloads of IoT applications can be handled easily in server-less computing environments, and this leads to cost-effective solutions. Users only pay for the compute time used by their functions, which is also a salient feature of cloud-like platforms.
- Server-less computing abstracts away the underlying infrastructure, allowing developers to focus on writing code for IoT applications rather than managing servers.
- Server-less computing enables real-time processing of IoT data, allowing for faster decision-making and response to IoT events.
- One important similarity between server-less computing and IoT is that both are event-driven architectures, making them a natural fit for each other.

Various issues, such as the number of information flows, data volume, and data protection are presently being handled by edge computing or fog computing [58,59]. Due to the low amount of available resources, this edge computing or fog computing can be transformed into server-less computing providing the basic idea of FaaS [54,60].

As is widely known, IoT is the interconnection of intelligent objects or devices using the Internet [61]. These devices having extensive human physiology detection capabilities are used in healthcare systems for remote health monitoring, diagnosis, and age-group-specific care [62,63]. So, opting for a server-less computing approach to support

proper workload and resource optimization in IoT-enabled healthcare systems is a wise choice. We can see many available services provided by various organizations, such as, AWS IoT Core (https://aws.amazon.com/iot-core/, accessed on 15 February 2024), Google Cloud IoT (https://cloud.google.com/iot-core, 15 February 2024), Azure IoT (https://azure.microsoft.com/en-us/products/iot-hub/, 15 February 2024), server-less IoT with AWS and NodeMCU (https://aws.amazon.com/blogs/compute/building-an-aws-iot-core-device-using-aws-server-less-and-an-esp32/, 15 February 2024), server-less IoT (https://kvaes.wordpress.com/2016/12/17/azure-iot-from-raspberrypi-with-sensor-to-azure-storage-table-by-using-a-server-less-architecture/, 15 February 2024) with Azure and Raspberry Pi etc.

5.2. Managing Medical Records with High Security

In a server-less platform, to manage various sensitive records like medical data can be stored using blockchain technology. Having a decentralized, immutable, auditable, and traceable data integrity blockchain technology can provide adequate security which is the need of the hour, ref. [64] specifically for medical records. Gill [65] has designed a model comprising modern technologies to provide the storing and managing of records with high security in a server-less environment. As shown in Figure 5, three layers are stacked and these are the service layer, management layer, and application layer.

Managing medical records with high security enhanced the clients' functionality. There is always a possibility of dealing with confidential data and users do not want to share it. These data can be analyzed and encrypted or anonymity at the client's end. These arrangements can be made in a server-less computing environment. It can be found that while scaling the data collection process in a cloud environment and authoring these data is not properly equipped and secured. While scaling, security, and trust services need to be transferred well across various cloud instances. There is a need for a cloud-secured gateway to achieve these approaches. As per the literature, to keep medical records with high security with the help of a server-less computing environment, we need to follow a few necessary considerations to ensure the confidentiality, integrity, and availability of sensitive patient information [1,66,67].

- We need to provide standard encryption services to sensitive medical data, while transmitting and storing.
- Controlled access provisions need to be in place so that only authorized personnel can access the data. Use Role-Based Access Control (RBAC) (https://www.techtarget.com/searchsecurity/definition/role-based-access-control-RBAC, accessed on 19 February 2024) and least privilege principles.
- APIs need to be secured enough, while accessing the sensitive data, ensuring the authenticity (https://cloud.google.com/healthcare-api/docs, accessed on 19 February 2024).
- The user logging information needs to be audited properly. We need to use proper monitoring tools to keep track of all activities and to detect unauthorized and abnormal activity.
- We need to ensure all regulatory standards have been complied with, such as HIPAA (https://aws.amazon.com/compliance/hipaa-compliance/, accessed on 19 February 2024) (Health Insurance Portability and Accountability Act) in the United States or GDPR (General Data Protection Regulation) in the European Union.
- We need to avoid storing redundant data, and unnecessary and outdated data.
- Proper backup and data recovery policies need to be adopted ensuring data availability.
- While developing any applications in server-less computing environments with sensitive medical data, we need to follow proper secured development practices, minimizing the vulnerabilities in application code and infrastructure configurations (https://csrc.nist.gov/projects/ssdf, accessed on 19 February 2024).

- Third-party services or libraries need to ensure that they meet security requirements (https://www.microsoft.com/en-in/industry/health/microsoft-cloud-for-healthcare?rtc=1, accessed on 19 February 2024).

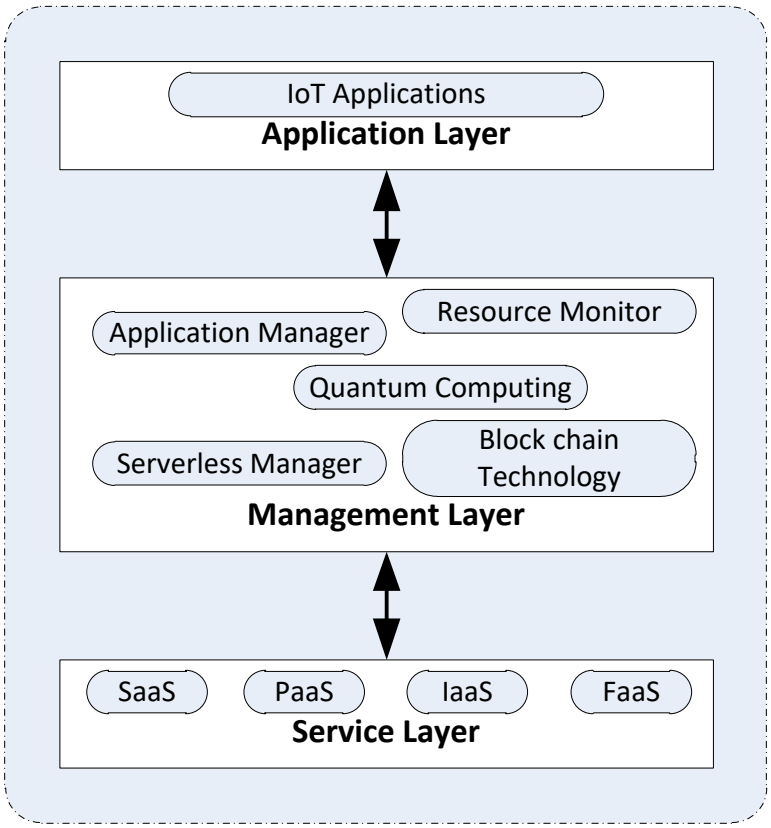

Figure 5. A model proposed by Gill [65].

5.3. Healthcare, Server-Less Computing, and Cloud Computing

Compared with traditional cloud computing, server-less provides added benefits like on-demand resources, nullifying the need for upfront commitment to utilizing the clients, and pricing as per short-duration execution of the client's business logic. Articles pages, blogs, forums, and many research works have pointed out the use of the server-less paradigm for commercial enterprise, security, transportation, education, etc. [68,69]. Cloud servers generally store and process the data to extract the health behavior or pattern which can used for personal care of that personal in real time [48,49,70–72].

Server-less computing is seen as more flexible than Platform-as-a-Service (PaaS) and less stringent than Software-as-a-Service (SaaS) [73]. A healthcare service is a value-based service, and that is why the collection of data using various ways is necessary. Cost needs to be optimal for these services to an increasing number of users. As the industry revolution is happening quickly, the requirements for digitization in the healthcare sector are becoming the need of the hour day by day. Thus, healthcare systems can reap the benefits of security, reliability, and accountability through server-less computing frameworks. There are several ways in which healthcare, server-less computing, and cloud computing are interconnected:

- Cloud computing can provide various solutions for healthcare data. Server-less can be used for data analytics, such as data transformation and analysis, without worrying about underlying server architecture.
- To deploy healthcare applications quickly and cost-effectively, a server-less computing environment is a nice choice. Cloud provides the necessary infrastructure ensuring scalability and reliability.

- To improve patient care we need to process the vast amount of medical data swiftly, and efficiently. IoMT (Internet of Medical Things) devices generate vast amounts of data that can be processed and analyzed using server-less computing and stored securely in the cloud.
- Cloud computing enables the delivery of telemedicine services, allowing healthcare providers to interact with patients remotely. Server-less computing can be used to develop and deploy telemedicine applications quickly and efficiently.
- Cloud computing providers offer compliance certifications and security features that are essential for handling sensitive healthcare data. Server-less computing can enhance security by reducing the attack surface area and providing built-in security features.
- Server-less computing's pay-as-you-go pricing model can help healthcare organizations reduce costs by only paying for the computing resources they use.
- Cloud computing also offers cost-effective storage and computing solutions compared to traditional on-premises infrastructure.

5.4. Server-Less Fog Computing

Fog computing extends cloud computing closer to the edge of the network, enabling faster data processing and reduced latency by leveraging resources such as edge devices, gateways, and local servers. Though we can find various existing frameworks for fog computing [74–77], programmability for modern IoT-based services is limited.

On the other hand, server-less computing provides computing in a lightweight, dynamic, and event-driven manner, which is a good fit for fog computing. This approach allows developers to build and deploy applications without having to manage the underlying infrastructure, while also benefiting from the low latency and proximity to data sources that fog computing offers.

Nowadays many services are connected to vehicles, drones, and cameras. That is why it is becoming very important to process the data in the edge network rather than doing everything in the cloud. These edge networks are close to both producers and customers, which is essentially what we can call fog computing. Data-centric IoT services can provide support for fog computing-based server-less frameworks. Although server-less computing deals with the execution rather than data management, this can create problems in data-intensive batch processing and data streaming activities. Server-less fog computing can be particularly useful in scenarios where real-time data processing and low latency are critical, such as in IoT (Internet of Things) applications, industrial automation, and autonomous vehicles. By combining the scalability and cost-effectiveness of server-less computing with the low latency and proximity to data sources of fog computing, organizations can build more efficient and responsive edge applications. A few key aspects of server-less fog computing have been listed below.

- Fog computing provides the advantage of data processing closer to the edge, and this is very much essential for IoT and industrial automation.
- As we know, server-less computing provides scalability on demand. In fog computing, the same concepts can be extended, enabling edge devices to scale resources based on the requirements of various applications.
- Server-less architectures abstract the underlying infrastructure, allowing developers to focus on writing code without managing servers. In Fog Computing, this leads to efficient utilization of edge resources, as these are allocated dynamically based on application needs.
- Challenges:
  - In fog computing, there will always be fewer resources for the edge devices, such as limited processing power, memory, and storage.
  - Optimizing data management to run efficiently on edge devices is a challenge.
  - Ensuring data consistency and reliability in a distributed environment is a challenge.
  - Securing edge devices and communication among devices and the cloud is crucial.

- Edge devices are more vulnerable to physical attacks and require robust security measures.
- There is always a chance of having a heterogeneous nature in an edge environment, and varied types of edge devices. It is a challenge to maintain interoperability of operations in this kind of heterogeneous environment.

6. Comparable Factors

Cost Analysis: Server-less computing has a pay-as-you-go business model. Companies have reported reduced capital expenses, as they no longer invested in servers. Traditional platforms had a higher expenditure for maintenance and incurred huge expenses on initial hardware [20].

User Feedback: Server-less Computing has been a reliable and accessible solution for healthcare management systems. End users and Healthcare professionals have positively commented on the speed and reliability of server-less platforms. raditionally everything was maintained by an administrator and frequent complaints of server slowdown were common [78].

Performance Metrics: Versatility and adaptability were demonstrated by healthcare apps that used server-less architecture, guaranteeing good performance regardless of traffic surges. Server-less design demonstrated quicker response times as well as lower downtime. Traditional platforms struggled as efficiency and performance were affected by heavy loads which resulted in higher response times [79].

Programming language: One important aspect of a server-less framework is, that it allows the programmer to write and execute the code in diverse programming languages. The whole computing environment can have a large number of APIs written in different languages [80].

Composability: Server-less computing offers the idea of building the application by writing the code in a modular manner without worrying about composability. Programmers can write functions independently for different types of business logic. server-less computing framework provides different ways of composability.

Deployment: Deployment of server-less applications in a cloud environment is very easy and easy to integrate continuously [81].

Security and Accounting: Security will be always a major concern in any kind of framework for computing. server-less computing environments separate the individual working environment significantly based on resource consumption, accounting information, and communication [78].

Scalability: In general, it is expected that cloud environments always provide better scalability. So in that same perspective, users will always expect the same from a server-less computing environment.

Monitoring and Debugging: Using a proper log file to monitor and debug the error frequently and periodically is an essential requirement to identify issues, errors, and a better understanding of the environment [82].

7. Issues and Challenges

To maintain and sustain a smart healthcare system in the server-less continuum, a developer might face a lot of issues and challenges. After going through various existing literature, we have enlisted a few. Although these are not all, it should provide a nice heads-up for the researcher and readers.

- One of the major challenges is to apply the data-oriented approach in fog computing with a server-less environment. Nowadays, IoT-based edge devices keep running various set modules and settings are fixed most of the time after deployment. In the healthcare domain, the sensor devices might have different situation-aware service modules. These service modules need to be triggered dynamically at the network edge as per the availability and mobility of these devices.

- Another critical issue is to maintain the data routing path. Due to various technical reasons, like computing location changing for various task instances in the cloud, device mobility, changing service modules dynamically as per business requirements, etc., manual configuration of the data routing path becomes problematic.
- In the case of bio-informatics, the biggest challenge is the data sequencing process itself in a server-less computing environment. Execution of these processes requires reservation and optimal utilization of computing power and managing the virtual servers.
- To handle medical images using parallel programming, server-less environment needs to have the scalability property.
- Many applications gather the data at the edge of the network and do the local processing before uploading the meaningful information to the cloud to maintain privacy and take advantage of the seamless profit from the elasticity property of cloud infrastructure. This creates a workflow continuum and maintaining this in a server-less environment is a big task.
- Every smart device can be connected with everything, due to the existence of IoT. This throws a few major challenges to server-less computing such as heterogeneity of devices, a large number of connected active devices, a safe and reliable environment, energy efficiency, etc.
- Server-less computing can be the golden choice for short-time services. But it will be inappropriate for long-term services.
- Existing server-less computing approaches try to provide scheduling mechanisms for various instances of tasks. However, a few shortcomings arise like burstiness, varied execution time, statelessness, and use of a single core.
- To attain zero latency or perform near real-time expectations, extreme parallelism, and large-scale resource handling are the main requirements from the nowadays AI algorithms. To perform these types of functions in a server-less computing environment requires consistent performance and higher scalability.
- To run a medical app in reality, generally, it needs to meet lots of expectations to provide more satisfaction to the user. To sustain these in a server-less environment, a developer needs to quickly alter, update, or fix the app to meet the requirements without affecting the performance.

8. Discussion and Recommendations

8.1. Discussion

After going through existing literature we can enumerate a few discussion points below.

- Combining artificial intelligence and machine learning can provide better-personalized recommendations, predictions, and decision support. This can provide accurate diagnostics, proper design of treatment plans, and improved patient trust.
- To support modern AI and machine learning tools with a large pool of resources, a server-less environment is a new paradigm.
- Major data center-oriented private organizations like Google, Amazon, IBM, Microsoft, etc., have already released their server-less platform to support function-based services.
- Various data-oriented applications [83–86] like database [87], video analysis [88], and image analysis can take the help of server-less computing to obtain results by executing the function-to-function basis as per requirement. Still, it might face cold-start issues with unknown approaches or unrecognized data.
- As time progresses, the need for dynamic and stochastic workload balancing will be the focus point in the current research scenario.
- To increase the cluster performance, the incoming request for function execution needs to be predicted using the history.
- To establish the idea of a smart healthcare system using a server-less environment, one needs to improve interoperability in IoT platforms.

8.2. Recommendations

In Figure 6, we can see an ideal architecture for server-less computing for Health Services. Server-less computing, due to its adaptive scaling feature, can react to more clients with greater throughput and a quicker ARR over non-server-less technology. The computational load is developed inside the Health FaaS framework utilizing Apache JMeter and the UCI ML Repository database.

- Server-less computing requires resource provision and management in smart health applications.
- Server-less computing can run on various layers, like edge, fog, or cloud. But, at different layers, there will be different bottlenecks
- Having a proper cluster-bed optimal virtualization, migration can be implemented to support the server-less computing paradigm.
- Using blockchain, a higher level of security and reliability can be provided in server-less computing.
- To work out the idea of a smart health system, we need to consider the IoT paradigm.
- If we think about latency, the execution of the functions as per user demand needs to be closer geographically.

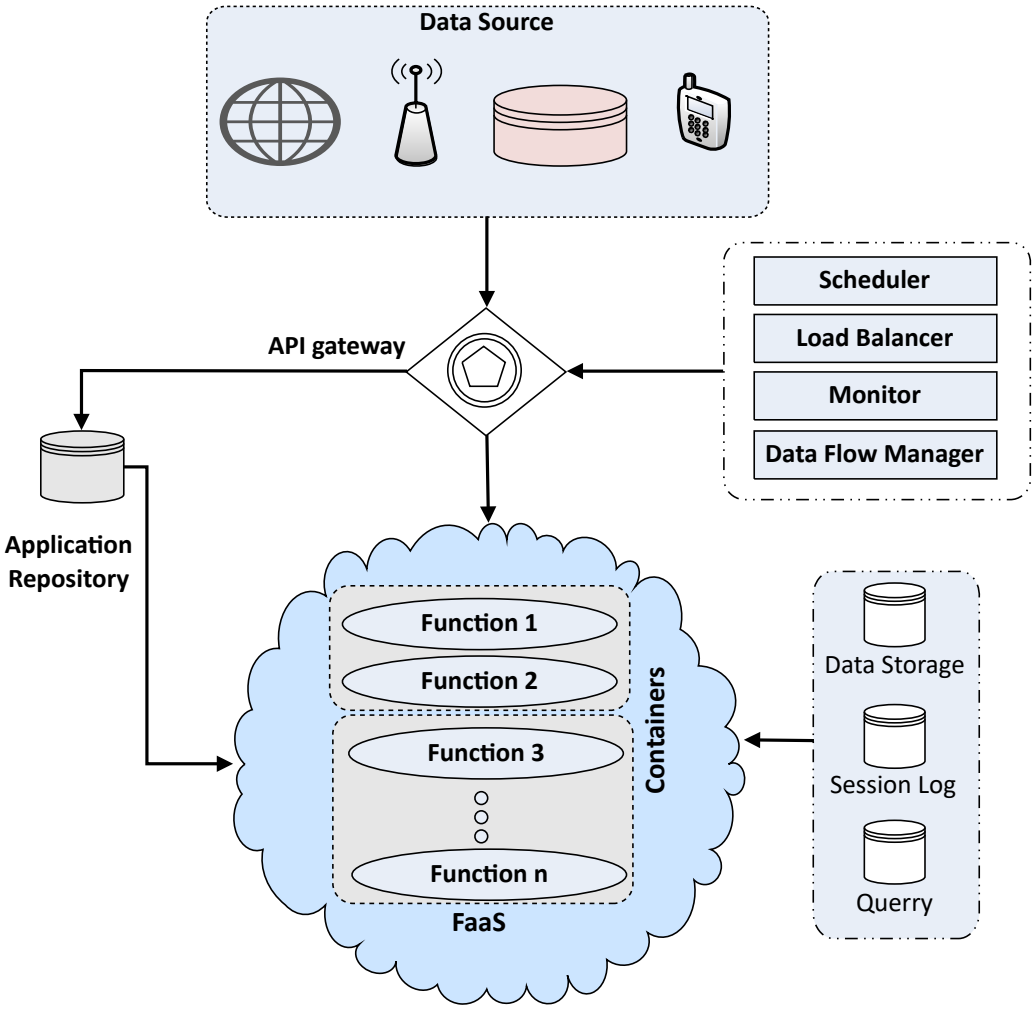

Figure 6. Ideal health FaaS server-less architecture.

9. Conclusions

Server-less computing has emerged as a promising solution for sustaining a digitized healthcare environment where multiple healthcare services can be deployed at a low scale which reduces costs and response times. Depending on customer requirements, server-

less providers can offer different types of services, including On-Demand, On-Reserve, or Instance Service. A server-less service model that can handle peak loads at any time is popular in the healthcare sector. Please note that different healthcare applications such as ECG, EEG, and electronic blood analysis require different pricing models. Healthcare systems such as remote monitoring systems and Fast Healthcare Interoperability Resources (FHIR) interfaces, often use services from various server-less vendors such as Amazon, IBM, and Microsoft. A server-less architecture is now being used as an efficient framework for delivering a variety of healthcare services. However, IoT and server-less integration in healthcare systems come with many challenges. Server-less frameworks are based on a set of independent stateless functions that cannot communicate with each other by exchanging data. This is one of the biggest problems with health management systems. Another challenge for server-less health management systems is maintaining trade-offs between cost and deadlines for various health applications. Despite its many challenges, it has emerged as a potential tool for handling the various services of the healthcare system. The identified research topic is considered one of the directions of research in the era of server-less frameworks in medical systems.

In this review paper, we have not provided experimental data regarding our proposed serverless healthcare framework as our empirical research studies are still ongoing. We are in the process of implementing the proposed methods and evaluating their outcomes in real-world test case scenarios. We aim to perform through investigations of these strategies in actual healthcare settings to validate their effectiveness and potential impact. Once these studies are complete, we will report our findings and discuss their implications.

Author Contributions: Conceptualization, R.K.D., R.K.B. and M.J.S.; methodology, R.K.D., R.K.B. and M.J.S.; software, A.G. and S.N.; validation, R.K.D., R.K.B. and M.J.S.; formal analysis, R.K.D.; investigation, M.J.S.; resources, R.K.B. and M.J.S.; data curation, R.K.D., A.G. and S.N.; writing—original draft preparation, R.K.D., A.G., S.N. and R.K.B.; writing—review and editing, M.J.S.; visualization, R.K.D., A.G. and M.J.S.; supervision, M.J.S.; project administration, M.J.S. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Data Availability Statement: Not Applicable.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Sbarski, P.; Kroonenburg, S. *Serverless Architectures on AWS: With Examples Using Aws Lambda*; Simon and Schuster: New York, NY, USA, 2017.
2. Figiela, K.; Gajek, A.; Zima, A.; Obrok, B.; Malawski, M. Performance evaluation of heterogeneous cloud functions. *Concurr. Comput. Pract. Exp.* **2018**, *30*, e4792. [CrossRef]
3. Sewak, M.; Singh, S. Winning in the era of serverless computing and function as a service. In Proceedings of the 2018 3rd International Conference for Convergence in Technology (I2CT), Pune, India, 6–8 April 2018; pp. 1–5.
4. Sampé, J.; Vernik, G.; Sánchez-Artigas, M.; García-López, P. Serverless data analytics in the IBM cloud. In Proceedings of the 19th International Middleware Conference Industry, Rennes, France, 10–14 December 2018; pp. 1–8.
5. Rosado, T.; Bernardino, J. An overview of openstack architecture. In Proceedings of the 18th International Database Engineering & Applications Symposium, Porto, Portugal, 7–9 July 2014; pp. 366–367.
6. Milojičić, D.; Llorente, I.M.; Montero, R.S. Opennebula: A cloud management tool. *IEEE Internet Comput.* **2011**, *15*, 11–14. [CrossRef]
7. Aditya, P.; Akkus, I.E.; Beck, A.; Chen, R.; Hilt, V.; Rimac, I.; Satzke, K.; Stein, M. Will serverless computing revolutionize NFV? *Proc. IEEE* **2019**, *107*, 667–678. [CrossRef]
8. Li, Z.; Guo, L.; Cheng, J.; Chen, Q.; He, B.; Guo, M. The serverless computing survey: A technical primer for design architecture. *ACM Comput. Surv. (CSUR)* **2022**, *54*, 1–34. [CrossRef]
9. Xavier, M.G.; Neves, M.V.; Rossi, F.D.; Ferreto, T.C.; Lange, T.; De Rose, C.A. Performance evaluation of container-based virtualization for high performance computing environments. In Proceedings of the 2013 21st Euromicro International Conference on Parallel, Distributed, and Network-Based Processing, Belfast, UK, 27 February–1 March 2013; pp. 233–240.
10. Agapito, G.; Cannataro, M. An overview on the challenges and limitations using cloud computing in healthcare corporations. *Big Data Cogn. Comput.* **2023**, *7*, 68. [CrossRef]

11. Osama, M.; Ateya, A.A.; Sayed, M.S.; Hammad, M.; Pławiak, P.; Abd El-Latif, A.A.; Elsayed, R.A. Internet of medical things and healthcare 4.0: Trends, requirements, challenges, and research directions. *Sensors* **2023**, *23*, 7435. [CrossRef]
12. Mishra, P.; Singh, G. Internet of medical things healthcare for sustainable smart cities: Current status and future prospects. *Appl. Sci.* **2023**, *13*, 8869. [CrossRef]
13. Kashyap, V.; Kumar, A.; Kumar, A.; Hu, Y.C. A systematic survey on fog and iot driven healthcare: Open challenges and research issues. *Electronics* **2022**, *11*, 2668. [CrossRef]
14. Dowdeswell, B.; Sinha, R.; Kuo, M.M.; Seet, B.C.; Hoseini, A.G.; Ghaffarianhoseini, A.; Sabit, H. Healthcare in Asymmetrically Smart Future Environments: Applications, Challenges and Open Problems. *Electronics* **2023**, *13*, 115. [CrossRef]
15. Restrepo, M.I.; Rousseau, L.M.; Vallée, J. Home healthcare integrated staffing and scheduling. *Omega* **2020**, *95*, 102057. [CrossRef]
16. Guido, R.; Solina, V.; Conforti, D. Offline patient admission scheduling problems. In Proceedings of the Optimization and Decision Science: Methodologies and Applications: ODS, Sorrento, Italy, 4–7 September 2017; Springer: Cham, Switzerland, 2017; pp. 129–137.
17. Ceschia, S.; Di Gaspero, L.; Schaerf, A. Simulated Annealing for the Home Healthcare Routing and Scheduling Problem. In Proceedings of the International Conference of the Italian Association for Artificial Intelligence, Udine, Italy, 28 November–2 December 2022; Springer: Cham, Switzerland, 2022; pp. 402–412.
18. Ngoo, C.M.; Goh, S.L.; Sze, S.N.; Sabar, N.R.; Abdullah, S.; Kendall, G.; A survey of the nurse rostering solution methodologies: The state-of-the-art and emerging trends. *IEEE Access* **2022**, *10*, 56504–56524. [CrossRef]
19. Iyengar, A.; Kundu, A.; Sharma, U.; Zhang, P. A trusted healthcare data analytics cloud platform. In Proceedings of the 2018 IEEE 38th International Conference on Distributed Computing Systems (ICDCS), Vienna, Austria, 2–6 July 2018; pp. 1238–1249.
20. Gadepalli, P.K.; Peach, G.; Cherkasova, L.; Aitken, R.; Parmer, G. Challenges and opportunities for efficient serverless computing at the edge. In Proceedings of the 2019 38th Symposium on Reliable Distributed Systems (SRDS), Lyon, France, 1–4 October 2019; pp. 261–2615.
21. Al-Roomi, M.; Al-Ebrahim, S.; Buqrais, S.; Ahmad, I. Cloud computing pricing models: A survey. *Int. J. Grid Distrib. Comput.* **2013**, *6*, 93–106. [CrossRef]
22. Khandelwal, A.; Kejariwal, A.; Ramasamy, K. Le taureau: Deconstructing the serverless landscape & a look forward. In Proceedings of the 2020 ACM SIGMOD International Conference on Management of Data, Portland, OR, USA, 14–19 June 2020; pp. 2641–2650.
23. Behera, R.K.; Rath, S.K. An efficient modularity based algorithm for community detection in social network. In Proceedings of the 2016 International Conference on Internet of Things and Applications (IOTA), Pune, India, 22–24 January 2016; pp. 162–167.
24. Kaewkasi, C. *Docker for Serverless Applications: Containerize and Orchestrate Functions Using OpenFaas, OpenWhisk, and Fn*; Packt Publishing Ltd.: Birmingham, UK, 2018.
25. Kumari, A.; Sahoo, B. Serverless architecture for healthcare management systems. In *Handbook of Research on Mathematical Modeling for Smart Healthcare Systems*; IGI Global: Hershey, PA, USA, 2022; pp. 203–227.
26. Shackelford, A. *Beginning Amazon Web Services with Node. js*; Apress: New York, NY, USA, 2015.
27. Mohan, A.; Sane, H.; Doshi, K.; Edupuganti, S.; Nayak, N.; Sukhomlinov, V. Agile cold starts for scalable serverless. In Proceedings of the 11th USENIX Workshop on Hot Topics in Cloud Computing (HotCloud 19), Renton, WA, USA, 8 July 2019.
28. Younge, A.J.; Von Laszewski, G.; Wang, L.; Lopez-Alarcon, S.; Carithers, W. Efficient resource management for cloud computing environments. In Proceedings of the International Conference on Green Computing, Chicago, IL, USA, 15–18 August 2010; pp. 357–364.
29. Zhang, M.; Zhu, Y.; Zhang, C.; Liu, J. Video processing with serverless computing: A measurement study. In Proceedings of the 29th ACM Workshop on Network and Operating Systems Support for Digital Audio and Video, Amherst, MA, USA, 21 June 2019; pp. 61–66.
30. Prechtl, M.; Lichtenthäler, R.; Wirtz, G. Investigating possibilites for protecting and hardening installable faas platforms. In Proceedings of the Service-Oriented Computing: 14th Symposium and Summer School on Service-Oriented Computing, SummerSOC 2020, Crete, Greece, 13–19 September 2020; Springer: Cham, Switzerland, 2020; pp. 107–126.
31. Swedha, K.; Dubey, T. Analysis of web authentication methods using Amazon web services. In Proceedings of the 2018 9th International Conference on Computing, Communication and Networking Technologies (ICCCNT), Bengaluru, India, 10–12 July 2018; pp. 1–6.
32. Sankaran, A.; Datta, P.; Bates, A. Workflow integration alleviates identity and access management in serverless computing. In Proceedings of the Annual Computer Security Applications Conference, Austin, TX, USA, 7–11 December 2020; pp. 496–509.
33. Kocher, P.; Horn, J.; Fogh, A.; Genkin, D.; Gruss, D.; Haas, W.; Hamburg, M.; Lipp, M.; Mangard, S.; Prescher, T.; et al. Spectre attacks: Exploiting speculative execution. *Commun. ACM* **2020**, *63*, 93–101. [CrossRef]
34. Lipp, M.; Schwarz, M.; Gruss, D.; Prescher, T.; Haas, W.; Mangard, S.; Kocher, P.; Genkin, D.; Yarom, Y.; Hamburg, M. Meltdown. *arXiv* **2018**, arXiv:1801.01207.
35. Brenner, S.; Kapitza, R. Trust more, serverless. In Proceedings of the 12th ACM International Conference on Systems and Storage, Haifa, Israel, 3–5 June 2019; pp. 33–43.
36. Jegan, D.S.; Wang, L.; Bhagat, S.; Ristenpart, T.; Swift, M. Guarding serverless applications with seclambda. *arXiv* **2020**, arXiv:2011.05322.

37. Alpernas, K.; Flanagan, C.; Fouladi, S.; Ryzhyk, L.; Sagiv, M.; Schmitz, T.; Winstein, K. Secure serverless computing using dynamic information flow control. *Proc. ACM Program. Lang.* **2018**, *2*, 1–26. [CrossRef]
38. Datta, P.; Kumar, P.; Morris, T.; Grace, M.; Rahmati, A.; Bates, A. Valve: Securing function workflows on serverless computing platforms. In Proceedings of the Web Conference 2020, Taipei, Taiwan, 20–24 April 2020; pp. 939–950.
39. Nastic, S.; Rausch, T.; Scekic, O.; Dustdar, S.; Gusev, M.; Koteska, B.; Kostoska, M.; Jakimovski, B.; Ristov, S.; Prodan, R. A serverless real-time data analytics platform for edge computing. *IEEE Internet Comput.* **2017**, *21*, 64–71. [CrossRef]
40. Chinchole, S.; Kulkarni, A.; Matai, L.; Kotadiya, C. A real-time cloud-based messaging system for delivering medication to the rural areas. In Proceedings of the 2017 International Conference on Intelligent Sustainable Systems (ICISS), Palladam, India, 7–8 December 2017; pp. 475–479.
41. Al-Masri, E.; Diabate, I.; Jain, R.; Lam, M.H.; Nathala, S.R. Recycle. io: An IoT-enabled framework for urban waste management. In Proceedings of the 2018 IEEE International Conference on Big Data (Big Data), Seattle, WA, USA, 10–13 December 2018; pp. 5285–5287.
42. Ergüzen, A.; Ünver, M. Developing a file system structure to solve healthy big data storage and archiving problems using a distributed file system. *Appl. Sci.* **2018**, *8*, 913. [CrossRef]
43. Niu, X.; Kumanov, D.; Hung, L.H.; Lloyd, W.; Yeung, K.Y. Leveraging serverless computing to improve performance for sequence comparison. In Proceedings of the 10th ACM International Conference on Bioinformatics, Computational Biology and Health Informatics, Niagara Falls, NY, USA, 7–10 September 2019; pp. 683–687.
44. Crespo-Cepeda, R.; Agapito, G.; Vazquez-Poletti, J.L.; Cannataro, M. Challenges and opportunities of amazon serverless lambda services in bioinformatics. In Proceedings of the 10th ACM International Conference on Bioinformatics, Computational Biology and Health Informatics, Niagara Falls, NY, USA, 7–10 September 2019; pp. 663–668.
45. Pérez, A.; Moltó, G.; Caballer, M.; Calatrava, A. A programming model and middleware for high throughput serverless computing applications. In Proceedings of the 34th ACM/SIGAPP Symposium on Applied Computing, Limassol, Cyprus, 8–12 April 2019; pp. 106–113.
46. Marefat, M.; Juneja, A. Serverless data parallelization for training and retraining of deep learning architecture in patient-specific arrhythmia detection. In Proceedings of the 2019 IEEE EMBS International Conference on Biomedical & Health Informatics (BHI), Chicago, IL, USA, 19–22 May 2019; pp. 1–4.
47. Paul, P.C.; Loane, J.; McCaffery, F.; Regan, G. A serverless architecture for wireless body area network applications. In Proceedings of the Model-Based Safety and Assessment: 6th International Symposium, IMBSA 2019, Thessaloniki, Greece, 16–18 October 2019; Proceedings 6; Springer: Cham, Switzerland, 2019; pp. 239–254.
48. Cheng, B.; Fuerst, J.; Solmaz, G.; Sanada, T. Fog function: Serverless fog computing for data intensive iot services. In Proceedings of the 2019 IEEE International Conference on Services Computing (SCC), Milan, Italy, 8–13 July 2019; pp. 28–35.
49. Kaffes, K.; Yadwadkar, N.J.; Kozyrakis, C. Centralized core-granular scheduling for serverless functions. In Proceedings of the ACM Symposium on Cloud Computing, Santa Cruz, CA, USA, 20–23 November 2019; pp. 158–164.
50. Eapen, B.R.; Sartipi, K.; Archer, N. Serverless on FHIR: Deploying machine learning models for healthcare on the cloud. *arXiv* **2020**, arXiv:2006.04748.
51. Pandey, S.R.; Hicks, D.; Goyal, A.; Gaurav, D.; Tiwari, S.M. Mobile notification system for blood pressure and heartbeat anomaly detection. *J. Web Eng.* **2020**, *19*, 747–774. [CrossRef]
52. Trilles, S.; González-Pérez, A.; Huerta, J. An IoT platform based on microservices and serverless paradigms for smart farming purposes. *Sensors* **2020**, *20*, 2418. [CrossRef] [PubMed]
53. Grzesik, P.; Mrozek, D. Serverless nanopore basecalling with AWS Lambda. In Proceedings of the Computational Science–ICCS 2021: 21st International Conference, Krakow, Poland, 16–18 June 2021; Proceedings, Part II 21; Springer: Cham, Switzerland, 2021; pp. 578–586.
54. Benedetti, P.; Femminella, M.; Reali, G.; Steenhaut, K. Experimental analysis of the application of serverless computing to IoT platforms. *Sensors* **2021**, *21*, 928. [CrossRef] [PubMed]
55. He, H.; Fu, S.; Wang, L.; Liu, S.; Wen, A.; Liu, H. MedTator: A serverless annotation tool for corpus development. *Bioinformatics* **2022**, *38*, 1776–1778. [CrossRef] [PubMed]
56. Grzesik, P.; Augustyn, D.R.; Wyciślik, Ł.; Mrozek, D. Serverless computing in omics data analysis and integration. *Briefings Bioinform.* **2022**, *23*, bbab349. [CrossRef] [PubMed]
57. Sadek, J.; Craig, D.; Trenell, M. Design and implementation of medical searching system based on microservices and serverless architectures. *Procedia Comput. Sci.* **2022**, *196*, 615–622. [CrossRef]
58. Yu, W.; Liang, F.; He, X.; Hatcher, W.G.; Lu, C.; Lin, J.; Yang, X. A survey on the edge computing for the Internet of Things. *IEEE Access* **2017**, *6*, 6900–6919. [CrossRef]
59. Xue, H.; Huang, B.; Qin, M.; Zhou, H.; Yang, H. Edge computing for internet of things: A survey. In Proceedings of the 2020 International Conferences on Internet of Things (iThings) and IEEE Green Computing and Communications (GreenCom) and IEEE Cyber, Physical and Social Computing (CPSCom) and IEEE Smart Data (SmartData) and IEEE Congress on Cybermatics (Cybermatics), Rhodes, Greece, 2–6 November 2020; pp. 755–760.
60. Elkholy, M.; Marzok, M.A. Light weight serverless computing at fog nodes for internet of things systems. *Indones. J. Electr. Eng. Comput. Sci.* **2022**, *26*, 394–403. [CrossRef]

61. Lu, Y.; Papagiannidis, S.; Alamanos, E. Internet of Things: A systematic review of the business literature from the user and organisational perspectives. *Technol. Forecast. Soc. Chang.* **2018**, *136*, 285–297. [CrossRef]
62. Nweke, H.F.; Teh, Y.W.; Mujtaba, G.; Al-Garadi, M.A. Data fusion and multiple classifier systems for human activity detection and health monitoring: Review and open research directions. *Inf. Fusion* **2019**, *46*, 147–170. [CrossRef]
63. Dey, N.; Ashour, A.S.; Bhatt, C. Internet of things driven connected healthcare. In *Internet of Things and Big Data Technologies for Next Generation Healthcare*; Springer: Cham, Switzerland, 2017; pp. 3–12.
64. Zheng, Z.; Xie, S.; Dai, H.; Chen, X.; Wang, H. An overview of blockchain technology: Architecture, consensus, and future trends. In Proceedings of the 2017 IEEE International Congress on Big Data (BigData Congress), Honolulu, HI, USA, 25–30 June 2017; pp. 557–564.
65. Gill, S.S. Quantum and blockchain based Serverless edge computing: A vision, model, new trends and future directions. *Internet Technol. Lett.* **2021**, *7*, e275. [CrossRef]
66. Baird, A.; Buliani, S.; Nagrani, V.; Nair, A. *AWS Serverless Multi-Tier Architectures; Using Amazon API Gateway and AWS Lambda*; Amazon Web Services Inc.: Seattle, WA, USA, 2015.
67. Podjarny, G.; Tal, L. *Serverless Security*; O'Reilly Media, Incorporated: Sebastopol, CA, USA, 2019.
68. Behera, R.K.; Sahoo, K.S.; Naik, D.; Rath, S.K.; Sahoo, B. Structural mining for link prediction using various machine learning algorithms. *Int. J. Soc. Ecol. Sustain. Dev. (IJSESD)* **2021**, *12*, 66–78. [CrossRef]
69. Armbrust, M.; Fox, A.; Griffith, R.; Joseph, A.D.; Katz, R.H.; Konwinski, A.; Lee, G.; Patterson, D.A.; Rabkin, A.; Stoica, I.; et al. *Above the Clouds: A Berkeley View of Cloud Computing*; Technical Report UCB/EECS-2009-28; EECS Department, University of California: Berkeley, CA, USA, 2009; Volume 28, p. 2009.
70. Gan, Y.; Zhang, Y.; Cheng, D.; Shetty, A.; Rathi, P.; Katarki, N.; Bruno, A.; Hu, J.; Ritchken, B.; Jackson, B.; et al. An open-source benchmark suite for microservices and their hardware-software implications for cloud & edge systems. In Proceedings of the Twenty-Fourth International Conference on Architectural Support for Programming Languages and Operating Systems, Providence, RI, USA, 13–17 April 2019; pp. 3–18.
71. Gunasekaran, J.R.; Mishra, C.S.; Thinakaran, P.; Kandemir, M.T.; Das, C.R. Implications of public cloud resource heterogeneity for inference serving. In Proceedings of the 2020 Sixth International Workshop on Serverless Computing, Delft, The Netherlands, 7–11 December 2020; pp. 7–12.
72. Huang, Z.; Mi, Z.; Hua, Z. HCloud: A trusted JointCloud serverless platform for IoT systems with blockchain. *China Commun.* **2020**, *17*, 1–10. [CrossRef]
73. Cusumano, M. Cloud computing and SaaS as new computing platforms. *Commun. ACM* **2010**, *53*, 27–29. [CrossRef]
74. Ananthanarayanan, G.; Bahl, P.; Bodík, P.; Chintalapudi, K.; Philipose, M.; Ravindranath, L.; Sinha, S. Real-time video analytics: The killer app for edge computing. *Computer* **2017**, *50*, 58–67. [CrossRef]
75. Jonas, E.; Schleier-Smith, J.; Sreekanti, V.; Tsai, C.C.; Khandelwal, A.; Pu, Q.; Shankar, V.; Carreira, J.; Krauth, K.; Yadwadkar, N.; et al. Cloud programming simplified: A berkeley view on serverless computing. *arXiv* **2019**, arXiv:1902.03383.
76. Wang, L.; Li, M.; Zhang, Y.; Ristenpart, T.; Swift, M. Peeking behind the curtains of serverless platforms. In Proceedings of the 2018 USENIX Annual Technical Conference (USENIX ATC 18), Boston, MA, USA, 11–13 July 2018; pp. 133–146.
77. Hellerstein, J.M.; Faleiro, J.; Gonzalez, J.E.; Schleier-Smith, J.; Sreekanti, V.; Tumanov, A.; Wu, C. Serverless computing: One step forward, two steps back. *arXiv* **2018**, arXiv:1812.03651.
78. Hong, S.; Srivastava, A.; Shambrook, W.; Dumitraș, T. Go serverless: Securing cloud via serverless design patterns. In Proceedings of the 10th USENIX Workshop on Hot Topics in Cloud Computing (HotCloud 18), Boston, MA, USA, 11–13 July 2018.
79. Bila, N.; Dettori, P.; Kanso, A.; Watanabe, Y.; Youssef, A. Leveraging the serverless architecture for securing linux containers. In Proceedings of the 2017 IEEE 37th International Conference on Distributed Computing Systems Workshops (ICDCSW), Atlanta, GA, USA, 5–8 June 2017; pp. 401–404.
80. Kumari, A.; Sahoo, B.; Behera, R.K.; Misra, S.; Sharma, M.M. Evaluation of integrated frameworks for optimizing qos in serverless computing. In Proceedings of the Computational Science and Its Applications–ICCSA 2021: 21st International Conference, Cagliari, Italy, 13–16 September 2021; Proceedings, Part VII 21; Springer: Cham, Switzerland, 2021; pp. 277–288.
81. Singh, C.; Gaba, N.S.; Kaur, M.; Kaur, B. Comparison of different CI/CD tools integrated with cloud platform. In Proceedings of the 2019 9th International Conference on Cloud Computing, Data Science & Engineering (Confluence), Noida, India, 10–11 January 2019; pp. 7–12.
82. Hendrickson, S.; Sturdevant, S.; Harter, T.; Venkataramani, V.; Arpaci-Dusseau, A.C.; Arpaci-Dusseau, R.H. Serverless computation with {OpenLambda}. In Proceedings of the 8th USENIX Workshop on Hot Topics in Cloud Computing (HotCloud 16), Denver, CO, USA, 20–21 June 2016.
83. Akkus, I.E.; Chen, R.; Rimac, I.; Stein, M.; Satzke, K.; Beck, A.; Aditya, P.; Hilt, V. {SAND}: Towards {High-Performance} serverless computing. In Proceedings of the 2018 Usenix Annual Technical Conference (USENIX ATC 18), Boston, MA, USA, 11–13 July 2018; pp. 923–935.
84. Oakes, E.; Yang, L.; Zhou, D.; Houck, K.; Harter, T.; Arpaci-Dusseau, A.; Arpaci-Dusseau, R. {SOCK}: Rapid task provisioning with {Serverless-Optimized} containers. In Proceedings of the 2018 USENIX Annual Technical Conference (USENIX ATC 18), Boston, MA, USA, 11–13 July 2018; pp. 57–70.
85. Jonas, E.; Pu, Q.; Venkataraman, S.; Stoica, I.; Recht, B. Occupy the cloud: Distributed computing for the 99%. In Proceedings of the 2017 Symposium on Cloud Computing, Santa Clara, CA, USA, 24–27 September 2017; pp. 445–451.

86. Ishakian, V.; Muthusamy, V.; Slominski, A. Serving deep learning models in a serverless platform. In Proceedings of the 2018 IEEE International Conference on Cloud Engineering (IC2E), Orlando, FL, USA, 17–20 April 2018; pp. 257–262.
87. Schleier-Smith, J. Serverless Foundations for Elastic Database Systems. In Proceedings of the CIDR, Asilomar, CA, USA, 13–16 January 2019.
88. Ao, L.; Izhikevich, L.; Voelker, G.M.; Porter, G. Sprocket: A serverless video processing framework. In Proceedings of the ACM Symposium on Cloud Computing, Carlsbad, CA, USA, 11–13 October 2018; pp. 263–274.