The concept of Privacy and Standardization of Microservice Architectures in cloud computing

R.Divya mounika[1], R .Naresh[2]

[1]M.Tech- Student, Department of Computer Science and Engineering, SRM Institute of Science and Technology, Kattankulathur, Chengalpattu, Chennai, Tamilnadu, India-603 203.
[2]Associate Professor, Department of Computer Science and Engineering, SRM Institute of Science and Technology, Kattankulathur, Chengalpattu, Chennai, Tamilnadu, India-603 203.*

E-Mail: rm2769@srmist.edu.in, nareshr@srmist.edu.in

Abstract: In monolithic systems replacement, microservices are being exploited as a natural solution. For improving the microservices in cloud technology, some specific technologies and standards have been considered, such as REST and API, on a massive scale for practical implementation. This paper is presented with a bibliographic survey on the architecture of microservice that focuses on privacy, security, and aspects of standardization in cloud computing environments. Based on microservices, different kinds of elements should be considered to determine the solutions effectively.

Keywords–API, Architecture, REST, Cloud, Microservice

1. INTRODUCTION

A significant issue is that the movement of monolithic architecture to the cloud. This paper researches the adopted microservices as a natural solution to replace the monolithic systems. The problem of security and privacy keys arises in a cloud computing system when using microservices' architecture. It was inspired by determining that much of the microservices have become a solution for cloud applications. A centralized pool of computing outsourcing mechanisms and configurable computing resources have been provided by cloud computing that allows different computing services to different people like utility-based systems like sewage, water, and electricity.

The microservices utilization has been addressed extensively in various applications owing to the diverse set of features in the architecture to be adopted for the cloud computing technologies with recent advances. At the threshold, cloud computing is implemented by users with low risk and cost. Cloud computing technology will become a demanding trend in any business application [1].

A lot of services are offered by monolithic services, and the most challenging issue is the scaling of monolithic applications. Some of them are popular, which require to be scaled as they are demanded highly. At the same time, the scaling of a complete set of services will also be made. It leads to considerable server resources by unpopular services, although they are not utilized [2].

For simplifying reality, the architecture has been emerged based on microservices, and it becomes an evolution naturally for application models.

Microservices include the following features [3] and refer to the software-oriented entity:
Based on a virtualized container, isolation can be done from the execution environment and other microservices.

Autonomy: Microservices are featuring independent deployment, movement, destroying, or duplication. Due to the capability of creating more than one instance of the same microservice, they couldn't be limited to any local resources.

All definite goals demonstrate using the available communication methods, either API or GUI by an open and standardized interface with effectiveness and efficiency.

Microservice is fine-grained: Each microservice should be responsible for dealing with its own task. The design pattern of cloud application is applied for the architecture of microservice that the classification into several small independent services and implements a specific feature, as mentioned in Figure 1.

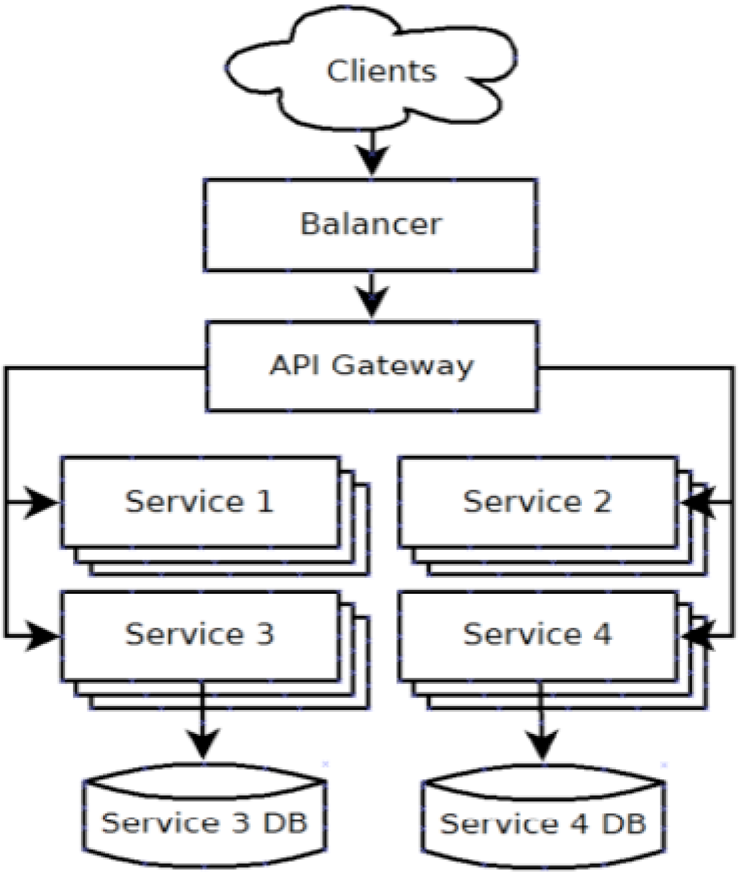

Figure 1. Microservice system architecture [3].

Because of the robust features such as communication with each other, independence, duplication, suspension, and moved to any computational resource, etc. However, meta-processes can be performed by microservices in a Meta operating system (OS) [13].

This paper includes sections as follows: The introduction of microservices architecture with implementation discusses in Section II and the security in the application of cloud computing presents in Section III. The privacy model in the cloud environment for microservices is demonstrated in Section IV. The cloud environment's standards show in Section V while Section VI concludes the microservice architecture.

2. MICROSERVICE ARCHITECTURE

In the software services industry, a dominant architectural style has been chosen by the microservice architecture. Microservice categorizes the system into small services that assist in processing a cohesive business function and the conventional style of service-oriented architecture with a natural evolution [4].

Application developers and service providers like eBay [9] [10], Netflix [7][8], Amazon [5][6] are showing much interest in using the concept of classifying the application into a set of interconnected and smaller services (microservices).

In the development of distributed applications, a pattern has been included in a microservice based architecture, which includes several smaller "independent" components with small applications [11].

As a microservice is typically a 3-tiered application [12], it comprises three layers such as a business logic layer [9], an interface layer [13], and a data persistence layer in the context of smaller bounded. The technical capabilities with a broad scope are set out a microservice could possess. Every microservice doesn't have all abilities. By relying on how the function is used, this would be varied. For instance, the API's providers primarily use a microservice with business logic, data persistence layers, and communications interface layer, but user interfaces have not been included essentially [11].

The microservices with a reference architecture model are considered in this project, demonstrating the elements and main components of the standard [11]. The microservice architecture and monolithic architecture are compared, which shows in Table 1.

Table 1. Comparing the Monolithic Architecture and Microservice Architecture

Category	**Monolithic Architecture**	**Microservice Architecture**
Code	A specific code base is applicable for complete architecture.	Each microservice has applied its own code base, i.e., multiple code bases.
Understandability	Confusing features most often that leads to difficulty in maintenance.	Easier maintenance and much better readability.
Deployment	The deployments is complex in nature with schedules downtimes and maintenance windows.	As each microservice can able to deploy individually, the deployment process is simple if not zero downtime.
Language	Only one programming language is used to develop the architecture.	Using different programming languages, the architecture of each microservice develops.
Scaling	Although the bottlenecks are localized, the scaling of entire application is required.	Without the requirement of entire application, the bottlenecked services can be

		scaled out.

The following elements will discuss in this paper:

A. API Proxy

Irrespective of the “API proxy” component, the proxy pattern implements at the level of microservice interface for “de-coupling” the microservice from its customers. API’s will be provided by organizations for different consumers, some outside and others within the enterprise. The microservices would be differentiated in different levels such as access levels, security requirements, service level agreements (SLA), etc. [11].

B. Enterprise API Registry

Based on the API registry service, the microservices’ “discovery” requirements are encountered. The microservice exposes the interfaces in such a way that visible for services’ consumers in both outsides and within the enterprise. A shared component across the enterprise is known as an “Enterprise API registry,” and its location should be accessible and well known. Based on a standard format, its content is published where the information must be in a human-readable format, consistent, and have controlled access. For viewing the information about API's available specifications at design time, the retrieval and search capabilities should have included in the Enterprise API Registry service [11].

C. Enterprise Microservice Repository

The “Enterprise Microservice Repository” is used to store the information about microservices. Accordingly, it has some specific details of this shared repository such as architecture, tools, the service it provides, lifecycle status of microservice, business and development ownership, versions, how it achieves the purpose, data persisted and queried, consuming any API’s, and any specific non-functional requirements. The standard specification artifacts for microservices should be defined by the enterprise when well-defined repository standards are not available [5].

These elements are considered as fundamental metrics to implement the microservices in an organized manner.

Ref. no	**Author name**	**Proposed work**	**Importance**
2	Mario Villamizar et al., (2015)	The microservice architecture pattern presents, assess, and test by the author. The large Internet companies like LinkedIn, Netflix, and Amazon are used it from the last years to deploy the cloud applications as a set of small services that can be operated, scaled, deployed, tested, developed, and upgraded independently. The cloud applications with the integration of microservice architecture	In deployment of scalable application efficiently, new opportunities provide by cloud computing that allows the enterprise applications for adjusting the computing resources on demand dynamically.

		pattern are allowed these companies to achieve the reduction of complexity, gaining of agility, and scaling of applications in the cloud efficiently.	
11.	Yale Yu et al., (2016)	Based on the definition of microservice concept and clear understanding, a microservice based reference architecture model presents by the author. In the enterprise architecture, building block and key architectural components are used for managing and implementing the enterprise microservices. While improving the solutions of enterprise IT transformation architecture, this model of reference architecture can give assistance for both IT professionals and business. At enterprise level, the microservice relevant technologies and API could be utilized properly without confusion.	When APIs and microservices leverage at enterprise level within an organization, a set of key architectural issues highlights and corresponding suggestions provide.
5.	A Krylovskiy et al., (2015)	As the Internet of Things (IoT) is identified as one of the prime enablers of the Smart City vision, it is being utilized in various application domains. It's a significant challenge to build the platforms of Smart City IoT with large-scale practically in despite of the wide adoption and standardization efforts of cloud computing applications and web standards. With the adoption of new technologies, these systems can able to evolve and scale in the dynamically changing IoT environments. In reverting to the challenges faced while establishing the Web distributed platforms and applications on a large-scale, microservice architecture has received much	For designing a platform of Smart City IoT, the architecture of microservice is utilized. By comparing with the approaches of generalized Service-Oriented Architecture (SOA), this architectural style is provided more benefits.

		popularity recently.	
12	Alam , et al., (2017)	Based on lightweight virtualization, a modular and scalable architecture protposes by the author. The proposed architecture offers the modularity that incorporates with the orchestration of lightweight virtualization facilitated by Docker which enables distributed deployments and simplifies management. With the distribution of application logic across various services where device failure or a single microservice couldn't impact on the performance of a system, the characteristics of fault-tolerance and availability are ensured. The instantiation and testing of proposed architecture is done based on a use case of simple time-dependent.	By using multiple commnication protocols, Industrial Internet of Things (IIoTs) depends on various devices working together, gathering, and sharing of information. In the development of architectures, a hindrance is resulted by this heterogeneity. The underlying protocols' operating applications can support by these architectures.
4	Nuha Alshuqayran et al., (2016)	The microservices architectures with a systematic mapping study and their implementation are presented. This investigation focuses on detecting the architectural challenges, quality attributes, and the architectural diagrams/views relevant to the microservice systems. For addressing the scalability and maintenance demands of online service providers, the microservices architecture is utilized. For summarising the progress and identifying the requirements and gaps, it's crucial that the microservice architecture is required for a study of systematic mapping as it is a new research area.	An increasing demand creates for microservice architecture through the acceleration of progress in the reliability, network security, and speed for moving services and software from being stored and processed locally on the machines of users to manage the third parties that could be accessible using the network.
6	Hamzeh Khazaei et al., (2016)	To investigate the microservice platforms' provisioning performance, a performance analytical model proposes and validates it using experiments.	Because of the manageability, scalability, flexibility, and performance, a new trend of microservice architecture has started for

		A microservice platform is designed and improved on the cloud of Amazon EC2 using Docker technology for detecting the essential elements that contributing to the microservice platforms performance. To establish a model of tractable analytical performance, the results and insights leverage from experiments. This model can use for processing capacity planning, and what-if analysis systematically for large scale microservices with minimum cost and amount of time.	application deployment/development in the cloud. To provide the complete cycle of software engineering for cloud applications, various platforms of microservice have become emerging services including from test, development, design, and deployment to maintenance.
7.	Robert Heinrich et al., (2017)	Various important technologies of deployment such as container orchestration, and container-based virtualization solutions have been emerged in addition to the architectural style. The cloud platforms can be exploited efficiently with the use of these technologies that provide a high degree of availability, scalability, and portability for microservices. A lack of performance is involved in the engineering approaches that consider the microservices particularities in despite of the giving importance to a sufficient level of performance.	New solutions are required for microservices complement approaches like DevOps and continuous delivery in terms of software architecture to improve the engineering performance. In regard to the monitoring, performance-aware testing, and modelling of microservices, open issues are detected and possible research directions are outlined.

3. SECURITY ON CLOUD COMPUTING

It’s essential to take some precautions while switching to a decentralized architecture from a centralized or monolithic architecture. In the previous period, security was responsible for receiving all service requests and focused on a single point [15]. Based on several access points that form a unique solution by interconnecting, the resources are provided in the microservice-based architecture.

By comparing with the microservices, the implementation of monolithic security services is more straightforward relatively. The intercommunications encapsulation and a clear boundary have been included in monolithic services. Within the system's inner layers, security vulnerabilities [16][17] will be obscured. A microservice also encapsulates the communications. The exact requirements are involved in both microservices and services.

The microservices require a simple routine completion in a microservice-based system for communicating with each other over a network. More information (endpoints) and data about the system will be exposed, and the attack surface is expanded [8]. In the same network, the communication between other services should be processed under consideration of some care, and this is one of the significant challenges [13] [15] [18] in this method.

The teams and services are generally subdivided by teams' organization for a system development based on microservices. However, these teams are used to implement and deliver services. For implementation, the alignment of teams and interconnection between them require in the microservices. For processing the communication, the protocol synchronization [19] is exploited. A standard is considered for improper interception or access protection. The essential point of security is that the definition of services is that they are interacting and interconnected [20].

Such network complexity is brought the security challenge, which creates a tough task in auditing, monitoring, debugging, and forensic analysis of the general application [21]. It’s a very tough task to establish a global view of the entire application as microservices deploy in a cloud that can’t control by the owners of an application [10].

A collection of workflows is essentially involved in an application based microservice architecture. Various levels of services can contain in these workflows. Before reaching out to the final destination, each one is processing and modifying the information. The certification of metadata relevant to a data stream and management of its validation during re-elaboration and time are required [22].

In a microservices architecture, a significant challenge is security, which should be considered carefully. A trust relationship is created through the communication of services with each other in different ways. For some systems, a user must be identified in all chains of service communication existed between microservices. For dealing with the security challenges [4], designers are employed well-known solutions such as OAuth and OAuth2.

The assurance of availability is the prime challenge in recent times [23], although the microservices do not cause dependencies among the modules and are independent. Currently, the DevOps movement (set of practices for integrating software development into IT operations) integrates with microservice architecture and cloud environments. From the code compilation, continuous integration is provided to the test availability and production environment. For systems development using microservices, a facilitator is made by it.

As the microservice architecture utilization facilitates a security requirement, ensuring the service availability is presented. In this approach, the entire solution is fragmented into smaller pieces [24]. In the event of a fragment failure, all system resources with unavailability wouldn’t result in this architecture by considering the fragments which are parts of code responsible to accomplish specific functions. As the microservices are bound to be observed, availability has crucial points such as invasions, software crash recovery, implementing software versions, and unavailability of infra features beyond points. Many instances need to start and stop over time and for the moments of running a particular service at any one time in a microservice architecture [25]. To provide the communication, locating the service providers by service consumers in real-time is to be enabled is service discovery [26].

Due to the ease of making the availability of new services and agility, Docker Containers have been required much hard work [23]. The packaging of microservices is allowed in the containers [27], and these are available for their dependencies in a single image, which leads to the minimization of downtime. Hence, this mode is termed code portability [28]. For service delivery, the docker containers have been utilized in the microservices that result in different benefits under various aspects. Those advantages are portability, independence, automation, and security, specifically while considering creating, easier management, and integration of environment systems continuously using the docker platform. In every container of Docker [28], the dependencies and application are required to run no less and no more are contained.

S.no	Author name	Proposed work	Importance
17	C Gadea et al., (2016)	By considering a scalable database of NoSQL, a reference architecture model presents that includes multiple subscribers to get notifications based on the “livequery” in case of any changes made instantly. This method provides how information from multiple disparate REST APIs can organize and transmit to the interested clients through the database by keeping the connection of WebSocket between an Object Synchronization Server and each client web browser.	By using backend-as-a-service providers and modern web frameworks, the real-time updates are reflected to a NoSQL data model in user interfaces applications of multiple subscribing end-user.
23	Kaibin bao et al., (2016)	Smart buildings and the Internet of Things (IoT) have been presented that identify in real buildings. To overcome these challenges, a decentralized service-oriented architecture has been improved based on a message-oriented middleware technology for the smart buildings field. To provide the applications composing from auxiliary services that offer maintainability, modularity, protocol adaption, and device abstraction, a network-transparent IoT message bus is used. By demonstrating how three different	For management of energy and Intranet of Things in smart buildings, a best-practice solution provides by the architecture. It is the main benefit of the proposed architecture than the conventional one.

		applications are used such as a generic user interface, automated building energy management, and privacy-enhancing energy data visualization, the flexibility of architecture is described.	
24	Daniel Escobar et al., (2016)	To modernize legacy applications into microservices, an approach is presented. To assess and visualize the current structure and dependencies between the data layer and the business layer, a model-centered process is utilized. By using four different diagrams, a modularization into clusters and microservices proposes furthermore. A JEE application with over 74566 LoC have been assess to verify the approach.	The challenge of maintenance and evolution of large applications could be faced by software development teams. This is termed as modernization and a process of understanding the existing application is contained. The first step towards the automated modernization of the application is constituted the proposed modularization.
26	Joe stubbs et al., (2015)	In microservice architectures, the challenges of service discovery and container technology and introduction of Serf node are introduced. By using Serf project, the designing of a fully decentralized open source solution is accomplished for the service discovery problem. One or more arbitrary Docker containers are contained in a Serf node which is a non-intrusive Docker image. The deployment of new images can be done into a cluster of Serf nodes in which the mechanisms of service discovery provides and advertises itself, self-healing, and monitoring are made.	Without any master node, the resulting cluster is a complete and homogeneous graph. Based on the extensibility of Serf node using Git, the solution for file system synchronization between Docker containers establishment is shown.
25	Razie Roostaei et al., (2017)	In the mobile cloud computing, a novel solution proposes for restricting issues	Due to the mobile devices' limited resources such as battery, storage, and

		based on movement of a user. To optimize the offloading decision, a mobility model and a fault tolerance mechanism are introduced. The offloading can be impacted by some parameters such as user movement, jitter, delay, and bandwidth.	computation, service disruptions can occur. To offload the computation and storage to the cloud, mobile devices can enable by the mobile cloud computing and the final results obtain from cloud servers.
27	David Jaramillo et al., (2016)	A lot of new complexity introduces by the approach of microservices architecture which needs the application developers who should have a considerable level of maturity for implementation of architectural style confidently. Docker is a disruptive technology that modifies the applications in such a way that they are being improved and distributed. To implement the architecture of microservices, Docker is a good choice with various benefits. In leveraging the architecture of microservices, proposed Docker can use effectively with a real working model as a case study.	Although microservices architecture is not highly demanded, it is gaining attention from companies who desire to market of a software product and to shorten time based on the improving of productivity effect and maximinzing the automation in the product’s lifecycle.

4. PRIVACY MODEL

In adopting cloud computing, privacy has become a barrier [24][29]. The migration to microservices has been used to restrict this issue as proposed scale gains in the architecture. Privacy is generally referred to as the state or condition of hiding the presence or view [30]. For privacy, or the confidential things require to be used, such as files and data. The user identity, data, and controls are needed to attain in cloud data storage privacy [31].

In large-scale scenarios of cloud computing, the sensitive information exchanging with several federations becomes powerful. To provide services, Service Providers (SP) and multiple Identity Providers (IdP) work collaboratively. For managing users' sensitive information, the models and privacy mechanisms should be provided by identity management [15].

For selecting the protection level required for the data, different options to the business customers are provided by cloud service. Encryption is the most common of these approaches. The customers choose the encryption type who can give preference and the encryption key is stored in a safe place under a monitoring system [19].

A well-referenced model is utilized to ensure privacy. In Figure 2, this model is shown.

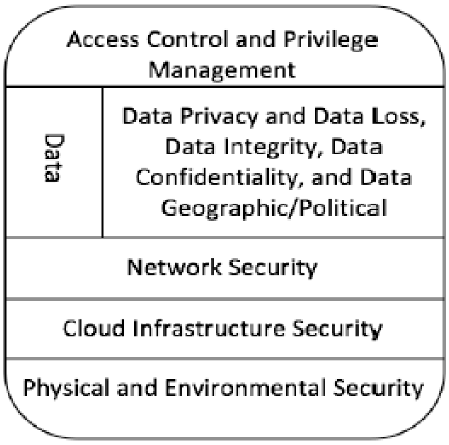
Access Control and Privilege Management
Data
Data Privacy and Data Loss, Data Integrity, Data Confidentiality, and Data Geographic/Political
Network Security
Cloud Infrastructure Security
Physical and Environmental Security

Figure 2. Model of cloud security and privacy [24].

Five layers are categorized from a secure and private cloud model based on the proposed model in [29]. They have included Privilege Management, Data and Access Control, Network Security, Cloud Infrastructure Security, and Physical and Environmental Security.

1. Physical and Environmental Security

The adoption of a layer of policies is done based on protection of physical access to the cloud provider [5].

2. Cloud Infrastructure Security

The addressing of issues is made to provide security for cloud infrastructure, particularly with virtualization [32].

3. Network Security

The medium is specified to which the end user is connected to the cloud that comprises the browsers and their connection [9].

4. Data Layer

Data Layer is covered integrity, data privacy, geographic location, and confidentiality [22].

5. Access Control and Privilege Management

The cloud services provider has used the policies and processes that ensure the utilization of granted appropriate privileges or modification of information for users. The issues of authorization [29], authentication [33], and identification are included in the cloud services provider.

Ref.no	**Author name**	**Proposed work**	**Importance**
29	C Perra (2016)	By integrating the user persuasion, performance metric	The services development and applications impacting

		computation and storage, and heterogeneous information sources, a framework proposes to develop the applications of sustainable urban mobility. The relevant mobility information, producing personalized information, and performing data processing are collected by a cloud application for sustainable urban mobility.	the sustainable mobility positively are fostered by modern information and communication technologies.
31	Haymanot Gebre-Amlak et al., (2017)	A mobility-inspired effective and efficient fog system with a design and prototype presents based on software-defined control over a mobile and wireless environment (MIST). The extension system termed as Fog Computing which is proposed recently that allows the computing in a smart device at network edge without allocating outsourcing responsibilities to a remote cloud. However, the cloud computing is complemented. For real-time urban surveillance tasks of uninterrupted target tracking, and dynamic big data driven, an approach of effective softwarization is investigated. To improve the awareness of a system, key technical challenges of node mobility are addressed.	Softwarization approaches have been focused on optimizing the costs and processes and provide new definitions of infrastructure and functional values in smart devices, services, M2M, networks, and storages. A new type of pervasive smart and mobile urban surveillance infrastructure is enabled with an advanced integration of mobile and wireless cyber physical systems that embed with smart sensors.
33	Z. Zhou et al., (2013)	For efficient privacy-aware data retrieval, a suit of novel techniques is presented. While moving insensitive information to public cloud, the basic concept is to retain the sensitive data in trusted private cloud and split the information. The current frameworks are not supported the privacy-aware data retrieval on hybrid cloud. The manual splitting of information has to be done by data owners. The most popular type MapReduce framework is	For economic savings and great flexibility, the data owner and advent of cloud computing is motivated for outsourcing the information to the cloud platform. Based on the concerns of data privacy, the development is getting troubled: the privacy data may include in the data owner and the information can’t be outsourced to cloud directly.

		adopted by the system known as Prometheus and the strategy of data partition is used that works independent to particular applications. The sensitive data can separate automatically from public data by Prometheus. Formally, the privacy-preserving feature of Prometheus has proved.	
30	Manfred Sneps-Sneppe et al., (2014)	The development of emerging telecom applications is in connection with the software standards. For the telecom providers (deployment and maintenance), and the developers (design and development), one of the major challenges presents by emerging (military) telecom applications. While transition from TDM to IP networks, it is true especially. The challenges associated with the priorities in security and transmission of data may bring by this transition.	For deployment and development of software standards, the research directions propose and the challenges presented by two biggest telecom related examples (FI-WARE and GIG).
33	Tran Quang Thanh et al., (2016)	An approach introduces by the author that allows the service providers and cloud application developers to consider the requirements of privacy and security across the lifecycle of an application. By considering various emerging methods such as Microservice Pattern Design and Network Functions Virtualization (NFV), a DevOps framework has been demonstrated specifically.	For supporting such direction, an application of proof-of-concept presents in the domain of healthcare.
28	K. El Makkaoui et al., (2016)	A new cloud security and privacy model (CSPM) provides that integrates into layers by considering the cloud providers during all phases of building and monitoring of cloud services. To restrict the barrier adoption of cloud services, this model will be used and to establish the confidence in cloud services and	To deliver the services over the Internet, cloud computing is an efficient solution increasingly and an adopted technology extensively as it provides diverse advantages such as high services of flexibility and scalability, reducing costs, services on demand,

		providing secure services. Based on the CSPM and some countermeasures, security attacks and threats are presented.	and configuration and sharing of computing resources.

5. MICROSERVICE STANDARDS AND SOLUTIONS

A natural way of standardization is involved in the centralized structure. But, this context is changing in the microservices implementation.

A different approach with standards is preferred by teams building microservices. For determining the solutions for similar issues to the ones that have been faced by other developers, the best choice is to use the context of producing useful tools instead of making use of a set of defined standards. These tools are shared with a broader group and gathered from implementations based on a git. The de-facto version control system is utilized the Github. In an in-house system, open-source practices are most common [34].

A microservice is a kind of application which processes the required functions. Independently, it has been evolved and can select its technology, architecture, and platform. The development methodology and release lifecycle can be deployed, managed, and scaled independently. The ESB and SOA construct is taken away by this approach, and the challenges are accompanied by making “smart endpoints.” The intermediate layers are referred to as the network resources which carry out the data transfer [11].

The Application Programming Interfaces (API) is defined as the applications that render interfaces that utilize other applications to interact with [5]. Based on the internet communication protocols like HTTP, microservice APIs are established that adhere to open standards such as SOAP [2], REST [35][36] and exploit data exchange technologies like JSON [5] and XML [18].

In a monolithic architecture, the improved applications carry out various functions such as customer credit check, product catalog, address validation, online ordering, etc. F these functions, applications are created if the pattern of microservice architecture is used. For providing the overall capability to the proposed service, these applications are cobbled together. The challenges of “monolithic” services and applications address application development using microservice architecture [11].

The implementation and documentation of microservices are mentioned below based on the researches.

A. Architectural views/diagrams [4]

- Standard modelling languages like YAML and RAML
- The modelling languages designed specifically, e.g. CAMLE
- Designed specification languages, e.g. Jolie
- Standard specification languages like JSON, Javascript (Node.js), and Ruby
- Psudocode for algorithms
- UML.

B. REST

A set of architectural principles is contained in the REpresentational State Transfer (REST), which creates a well-defined interface design. REST principles are used in applications known as RESTFul. For providing services to other services like web services, REST [10][18][36][37] is implemented often and to the use of the same messages entirely. It's essential to highlight three critical concepts for understanding the architectural style in a better way, such as (i) feature, (ii) operations and (iii) representations. Using a unique identifier (URI), information is available for consumers, i.e., resource, which can also be referred to as the source of representations. The state of the requested resource is demonstrated by a set of data in the representations. A defined hierarchy and a notation pattern should have included in URIs with descriptive nature. One or more URIs can identify the same resources, but only one resource is detected by a URI [38][39].

C.API

Application Program Interface (API) are

(a) Basic authentication with the inclusion of registering API user including the protection of strong password,
(b) Modern security mechanisms like web encryption, message-level security, and web encryption, and
(c) As a third security factor, the security mechanism is implemented within the API and its back-end services like transport layer handshake protocol, back-end authentication, and token-based API for critical public infrastructure [13].

In various technologies, REST APIs [7] have been improved, and microservices are developed based on various kinds of programming languages like NodeJs, Scala, Phyton, Ruby, PHP, .NET, Java, etc. and persistent technologies like NoSQL, SQL, etc. [2][28]. The web clients can access and manage the REST APIs. They can expose to the microservices and receive their responses using a mechanism of "livequery" in which the database data is updated and communicated to subscribing clients instantly [18]. To design the REST-based web services, the best categories of practices are presented in Figure 3.

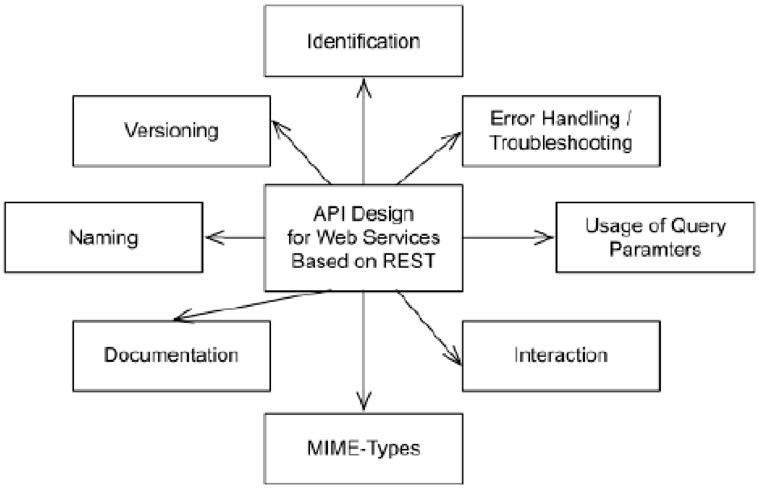

Figure 3. Presenting categories with best practices to design REST-based web services [40].

In these developments [18][41][42][43], NoSQL database are utilized. To provide the sharding, scaling, and replication functionality expected from modern architectures, it’s essential that the NoSQL nature of the database support required hierarchical information for document editing collaboratively [18].

According to the reports of a well-known jobs portal indeed.com, the architecture based on microservices popularity shows that the number of job openings has increased over 100 times from the last six years for microservices-related technologies. The identical growth of jobs has remained in similar technology areas like XML and SOAP [10].

In this section, the implementation of simple algorithms that should meet particular requirements is presented based on provided microservices solutions.

S. No	**Author name**	**Proposed working**	**Importance**
10	Y. Sun et al., (2015)	For applications of microservices-based cloud, a design based on security-as-a-service is proposed. An infrastructure of flexible monitoring and policy enforcement is established with the addition of a new API primitive FlowTap for the network hypervisor to secure the cloud applications for network traffic. Based on the deployment of Bro network monitor using FlowTap, the effectiveness of solution is demonstrated. To support different scenarios of monitoring and policies, the solution is flexible enough and incurred minimal overhead (~6%) for real world usage based on the analysed results.	The deployment, developing, and scaling of different parts of an application independently is allowed in microservice architecture. It becomes a trend for improving the cloud applications.
34	S. Yamamoto et al., (2012)	In the smart city, the data is taken from smart houses and a platform proposes known as Scallop4SC which helps to process and store the large-scale house data that categorizes into data or log data. Based on a multi-node cluster, the Hadoop/MapReduce intorudces as the log amount is large extremely. For managing the heterogeneous log data in a schemaless manner, HBase key-value store is used. MySQL is	To achieve value-added services, a wide-variety of data are gathered from devices and sensors in the smart city environment. By using 12 Linux servers, a prototype is implemented. An experimental evaluation is conducted to determine the energy consumption devise-wise based on the

		chosen for processing different demands to the house data and managing the configuration data efficiently.	recorded actual house log for one year in the smart house.
35	J. Bogner and A. Zimmermann (2016)	The methodologies and models of extending Enterprise Architecture (EA) propose by the author and they provide the information about a high degree of distribution and heterogeneity for supporting the digital transformation and relevant information systems with micro-granular architectures. For both the capabilities of business and IT within adaptable digital enterprise architecture, the flexibility and agile transformation are supported.	For supporting flexible and agile services and products in the digital transformation, IT environments have been emerged as they contain huge number of small structures such as mobility systems, microservices, and Internet of Things (IoT).
4.	Nuha Alshuqayran et al., (2016)	The microservices architecture with implementation is presented based on a systematic mapping study. The detection of challenges in the architecture and diagrams or view of architecture, and quality attributes relevant to the microservice systems is focused in this study. For addressing the demands of scalability and maintenance of online service providers, the architecture of microservices is used. It's essential that the systematic mapping study is required because the microservice architecture becomes an emerging technology that helps to identify the gaps and requirements and summarise the progress.	A higher demand is created by enhancing the progress of security, reliability, and network speed for moving services and software from the phase of storing and processing on machines of users to being the third parties management that can accessible over the network.
11.	Yale Yu et al., (2016)	To implement and manage the enterprise microservices in the an enterprise architecture, a microservice based reference architecture model presents based on the microservice definition with clear understanding and key architectural components. While developing the solutions of	The modern enterprise business is including the leading enabler and partner as ICT. The transformation of pure digital style and virtual organized enterprises is made using different companies. A group of

		enterprise IT transformation architecture, this reference architecture model can give guidance to both IT professionals and business as well. Hence, the microservice and API could exploit at enterprise level with proper implementation.	microservice based applications and modules with a business oriented structure is enabled and supported this digital transformation and they are new to most of the enterprises and companies.
17	C. Gadea et al., (2016)	Based on the concept of a scalable NoSQL database, a reference architecture presents that can allow multiple subscribers for receiving notifications of database immediately using the “livequery”. This paper displays how the information from multiple disparate REST APIs can organize and transmit to the interested clients using database while keeping one open connection of WebSocket between an Object Synchronization Server and client web browser.	For a NoSQL data model, backend-as-a-service providers and modern web frameworks have been utilized to receive real-time updates instantly to be provided in the applications of multiple subscribing end-user’s interfaces.
37.	P. Marchetta et al., (2015)	A core component of the S^2 is presented: The smart mobility services provide using designing and implementation of map-based web platform. The use cases and the platform’s design and implementation are presented. The efficiency and accuracy of the Map Matching and Traffic Monitoring algorithms are evaluated. Based on the simulations in SUMO, the proposed platform is implemented using the generated realistic urban traffic data.	A new vision of urban mobility is smart mobility. ICT and private and public transportation systems are required to develop the scenarios of smart mobility that has a deep integration among citizens. An architecture is proposed with the S^2 that can able to update and collect the data.

6. CONCLUSIONS

Microservice-based architecture becomes an emerging architectural style for software development. By constructing smaller software components, the microservice architecture develops that has numerous kinds of benefits than the conventional monolithic architecture such as ease of scaling, and increasing the resilience. This project presents the considerable elements for implementing the solutions using microservices, the detection of elements that implement in the cloud computing, and demonstrating the microservice architecture along with the integration of solutions and standards to the microservices architecture using privacy model.

7. REFERENCE:

[1]. J. Lin, L. Chaoyu, and S. Huang, “Migrating Web Applicationsto Clouds with Microservices Architectures,” *Int. Conf. Appl.Syst. Innov.*, pp. 1–4, 2016.

[2]. M. Villamizar and et al, “Evaluating the Monolithic and theMicroservice Architecture Pattern to Deploy Web Applications inthe Cloud Evaluando el Patrón de Arquitectura Monolítica y deMicro Servicios Para Desplegar Aplicaciones en la Nube,” *10thComput. Colomb. Conf.*, pp. 583–590, 2015.

[3]. D. I. Savchenko, G. I. Radchenko, and O. Taipale,“Microservices validation: Mjolnirr platform case study,” *201538th Int. Conv. Inf. Commun. Technol. Electron. Microelectron.MIPRO 2015 - Proc.*, no. May, pp. 235–240, 2015.

[4]. N. Alshuqayran, N. Ali, and R. Evans, “A systematic mappingstudy in microservice architecture,” *Proc. - 2016 IEEE 9th Int.Conf. Serv. Comput. Appl. SOCA 2016*, pp. 44–51, 2016.

[5]. A. Krylovskiy, M. Jahn, and E. Patti, “Designing a Smart CityInternet of Things Platform with Microservice Architecture,”*Proc. - 2015 Int. Conf. Futur. Internet Things Cloud, FiCloud2015 2015 Int. Conf. Open Big Data, OBD 2015*, pp. 25–30,2015.

[6]. H. Khazaei, C. Barna, N. Beigi-Mohammadi, and M. Litoiu,“Efficiency analysis of provisioning microservices,” *Proc. Int.Conf. Cloud Comput. Technol. Sci. CloudCom*, pp. 261–268,2017.

[7]. R. Heinrich *et al.*, “Performance Engineering for Microservices:Research Challenges and Directions,” *Proc. 8th ACM/SPEC Int.Conf. Perform. Eng. Companion*, pp. 223–226, 2017.

[8]. M. Ahmadvand and A. Ibrahim, “Requirements reconciliation forscalable and secure microservice (de)composition,” *Proc. – 2016IEEE 24th Int. Requir. Eng. Conf. Work. REW 2016*, pp. 68–73,2017.

[9]. T. Q. Thanh, S. Covaci, T. Magedanz, P. Gouvas, and A.Zafeiropoulos, “Embedding security and privacy into thedevelopment and operation of cloud applications and services,”*2016 17th Int. Telecommun. Netw. Strateg. Plan. Symp.*, pp. 31–36, 2016.

[10]. Y. Sun, S. Nanda, and T. Jaeger, “Security-as-a-service formicroservices-based cloud applications,” *Proc. - IEEE 7th Int.Conf. Cloud Comput. Technol. Sci. CloudCom 2015*, pp. 50–57,2016.

[11]. Yale Yu, H. Silveira, and M. Sundaram, “A microservice basedreference architecture model in the context of enterprisearchitecture,” *2016 IEEE Adv. Inf. Manag. Commun. Electron.Autom. Control Conf.*, pp. 1856–1860, 2016.

[12]. Alam, Muhammad, Joao Rufino, Joaquim Ferreira, Syed Hassan Ahmed, Nadir Shah, and Yuanfang Chen. "Orchestration of microservices for iot using docker and edge computing." *IEEE Communications Magazine* 56, no. 9 (2018): 118-123.

[13]. K. Bakshi, “Microservices-based software architecture andapproaches,” *IEEE Aerosp. Conf. Proc.*, 2017.

[14]. J. Werner, C. M. Westphall, and C. B. Westphall, “Cloud identitymanagement: A survey on privacy strategies,” *Comput.Networks*, vol. 122, pp. 29–42, 2017.

[15]. Giri, Shailendra, and Subarna Shakya. "Cloud Computing and Data Security Challenges: A Nepal Case." *International Journal of Engineering Trends and Technology, 67 (3), 146* 150 (2019).

[16]. Meneghello, Francesca, Matteo Calore, Daniel Zucchetto, Michele Polese, and Andrea Zanella. "IoT: Internet of Threats? A survey of practical security vulnerabilities in real IoT devices." *IEEE Internet of Things Journal* 6, no. 5 (2019): 8182-8201.

[17]. C. Gadea, M. Trifan, D. Ionescu, and B. Ionescu, "A referencearchitecture for real-time microservice API consumption," *Proc.3rd Work. CrossCloud Infrastructures Platforms – CrossCloud'16*, pp. 1–6, 2016.

[18]. S. Srinivasan, "Data privacy concerns involving cloud," *201611th Int. Conf. Internet Technol. Secur. Trans. ICITST 2016*, pp.53–56, 2017.

[19]. A. Ciuffoletti, "Automated Deployment of a Microservice-basedMonitoring Infrastructure," *Procedia Comput. Sci.*, vol. 68, pp.163–172, 2015.

[20]. M. Fazio, A. Celesti, R. Ranjan, C. Liu, L. Chen, and M. Villari,"Open Issues in Scheduling Microservices in the Cloud," *IEEECloud Comput.*, vol. 3, no. 5, pp. 81–88, 2016.

[21]. Melis, Andrea, Silvia Mirri, Catia Prandi, Marco Prandini, Paola Salomoni, and Franco Callegati. "Integrating personalized and accessible itineraries in MaaS ecosystems through microservices." *Mobile Networks and Applications* 23, no. 1 (2018): 167-176.

[22]. H. Kang, M. Le, and S. Tao, "Container and microservice drivendesign for cloud infrastructure DevOps," *Proc. - 2016 IEEE Int.Conf. Cloud Eng. IC2E 2016 Co-located with 1st IEEE Int. Conf.Internet-of-Things Des. Implementation, IoTDI 2016*, pp. 202–211, 2016.

[23]. K. Bao, I. Mauser, S. Kochanneck, H. Xu, and H. Schmeck, "AMicroservice Architecture for the Intranet of Things and Energyin Smart Buildings," *Proc. 1st Int. Work. Mashups Things APIs -MOTA '16, pp. 1–6, 2016.*

[24]. D. Escobar *et al.*, "Towards the understanding and evolution of monolithic applications as microservices," *Proc. 2016 42nd Lat.Am. Comput. Conf. CLEI 2016*, 2017.

[25]. R. Roostaei and Z. Movahedi, "Mobility and Context-Aware Offloading in Mobile Cloud Computing," Proc. - 13th IEEE Int. Conf. Ubiquitous Intell. Comput. 13th IEEE Int. Conf. Adv. Trust. Comput. 16th IEEE Int. Conf. Scalable Comput. Commun. IEEE Int., pp. 1144–1148, 2017.

[26]. J. Stubbs, W. Moreira, and R. Dooley, "Distributed Systems of Microservices Using Docker and Serfnode," Proc. - 7th Int. Work. Sci. Gateways, IWSG 2015, pp. 34–39, 2015.

[27]. D. Jaramillo, D. V. Nguyen, and R. Smart, "Leveraging microservices architecture by using Docker technology," *Conf.Proc. - IEEE SOUTHEASTCON*, vol. 2016–July, pp. 0–4, 2016.

[28]. K. El Makkaoui, A. Ezzati, A. Beni-Hssane, and C. Motamed, "Data confidentiality in the world of cloud," *J. Theor. Appl. Inf.Technol.*, vol. 84, no. 3, pp. 305–314, 2016.

[29]. C. Perra and S. Member, "A Framework for the Development of Sustainable Urban Mobility Applications," 2016.

[30]. Sneps-Sneppe, Manfred, and Dmitry Namiot. "Micro-service architecture for emerging telecom applications." *International Journal of Open Information Technologies* 2, no. 11 (2014): 34-38.

[31]. H. Gebre-amlak, S. Lee, A. M. A. Jabbari, Y. Chen, and B. Choi, "MIST : Mobility-Inspired SofTware-Defined Fog System," 2017.

[32]. R. H. Steinegger, D. Deckers, P. Giessler, and S. Abeck, "Riskbased authenticator for web applications," *Proc. 21st Eur. Conf.Pattern Lang. Programs - Eur. '16*, no. February 2017, pp. 1–11, 2016.

[33]. T. Q. Thanh, S. Covaci, T. Magedanz, P. Gouvas, and A. Zafeiropoulos, "Embedding security and privacy into the development and operation of cloud applications and services," 2016 17th Int. Telecommun. Netw. Strateg. Plan. Symp., pp. 31– 36, 2016

[34]. S. Yamamoto, S. Matsumoto, and M. Nakamura, "Using cloudtechnologies for large-scale house data in smart city," *CloudCom2012 - Proc. 2012 4th IEEE Int. Conf. Cloud Comput. Technol.Sci.*, pp. 141–148, 2012.

[35]. J. Bogner and A. Zimmermann, "Towards IntegratingMicroservices with Adaptable Enterprise Architecture," *Proc. -IEEE Int. Enterp. Distrib. Object Comput. Work. EDOCW*, vol.2016–Septe, pp. 158–163, 2016.

[36]. D. Guo, W. Wang, G. Zeng, and Z. Wei, "Microservicesarchitecture based cloudware deployment platform for servicecomputing," *Proc. - 2016 IEEE Symp. Serv. Syst. Eng. SOSE2016*, pp. 358–364, 2016.

[37]. P. Marchetta, E. Natale, A. Pescape, A. Salvi, and S. Santini, "Amap-based platform for smart mobility services," *Proc. – IEEESymp. Comput. Commun.*, vol. 2016–Febru, pp. 19–24, 2016.

[38]. A. de Camargo, I. Salvadori, R. dos S. Mello, and F. Siqueira,"An architecture to automate performance tests onmicroservices," *Proc. 18th Int. Conf. Inf. Integr. Web-basedAppl. Serv. - iiWAS '16*, pp. 422–429, 2016.

[39]. P. Giessler, R. Steinegger, S. Abeck, and M. Gebhart, "Checklistfor the API Design of Web Services based on REST," vol. 9, no.3, pp. 41–51, 2016.

[40]. A. Gueidi, H. Gharsellaoui, and S. Ben Ahmed, "A NoSQLbasedApproach for Real-Time Managing of Embedded DataBases," *Proc. - 2016 World Symp. Comput. Appl. Res. WSCAR2016*, pp. 110–115, 2016.

[41]. T. I. Damaiyanti, A. Imawan, and J. Kwon, "Extracting trends oftraffic congestion using a NoSQL database," *Proc. - 4th IEEEInt. Conf. Big Data Cloud Comput. BDCloud 2014 with 7th IEEEInt. Conf. Soc. Comput. Networking, Soc. 2014 4th Int. Conf. Sustain. Comput.C, pp. 209–213, 2015.*

[42]. R. Simmonds, P. Watson, and J. Halliday, "Antares: A Scalable,Real-Time, Fault Tolerant Data Store for Spatial Analysis," Proc. - 2015 IEEE World Congr. Serv. Serv. 2015,pp. 105–112, 2015.