Security Concerns regarding Software Development Migrations in Quantum Computing Context

Andreea-Alina Cornea, Alexandru Marius Obretin

Department of Informatics and Economic Cybernetics, Bucharest University of Economic Studies, 15-17 Calea Dorobanti, 010552, Bucharest, Romania

Abstract – Living in the world managed by technology involves taking into consideration all security aspects which may help, protect, and maintain digital environment healthy. As the technology evolves, the importance of upgrading is emphasized by customer perception and satisfaction and justifies the decision to switch from obsolete alternatives, as monolithic applications which can be found in most public system architectures developed during 2000s, to microservices and increased security ecosystems, as an indisputable requirement. Even so, current times are defined by innovation and quantum computers future usage seems to define a totally new era of security. This paper represents an overview of how this future perspective may affect public system applications and aims to provide a possible approach for integrating mathematical constructs into the security layer to increase protection for the modern software architectural style represented by microservices and which has become the favourite development alternative of the last years.

Keywords – microservices; encryption; quantum computers; lattice problems.

1. Introduction

Rapid development of modern technologies is impacting various facets of our lives. Modern technologies simplified processes, brought connectivity between people and resources, and provided proper tools for tackling challenging problems. Nowadays, the most valuable companies on Earth are technology companies that in a way or another outscored their competitors and acquired worldwide recognition. Either proposing novel business models that brought immediate success or adopting a steady incremental growth, they relied on digital innovation.

As a consequence, this technological revolution profoundly affected the old status quo, some jobs were replaced, others have been invented and society as a whole has been forced to adapt. But this digital world where everything is accessible from the comfort of one’s living room comes with great responsibility. The interconnected world not only allowed services to better approach customers, but also opened a playground for criminals. According to the 2019 Cost of Cybercrime Study released by Accenture, organizations spend more than ever to deal with the costs and consequences of more sophisticated attacks and “the average cost of cybercrime for an organization increased from 1.4 million to 13.0 million USD” [1]. According to the same report, best practices in the realm of cybercrime could generate up to 5.2 trillion USD globally over the next five years. For comparison, according to the World Bank [2], in 2018, Japan, the 3rd largest economy on the planet, had a gross domestic product of 4.971 trillion USD. The growing rate of cybercrimes should worry individuals and institutions alike. These days, more than ever, security in respect with information technology should not be taken for granted.

Public institutions need to quickly accommodate the challenges that modern technology development brings. Bearing in mind this assumption, there is an obvious requirement to reflect on the default way so
far by developing software applications, known in the specialty literature as monolith architecture. Understanding the fact that the monolithic architectural pattern is not anymore considered a suitable approach to follow by modern software development standards, the relevant institutions should determine, analyse and elaborate strategies in order to keep up with current best practices.

Moreover, concrete actions are compulsory to provide reliable and secure public service software solutions that simultaneously offer citizens confidence and decent user experiences in interaction with such platforms.

In the process of migrating an application from monolith to a modern architecture, some essential aspects, as security, should be taken into consideration. The case described in this paper is based on public system applications which need an increased attention when the security layer is designed. The public institutions have the social duty to investigate and provide the most secure alternatives in order to assure the protection and integrity of civilians’ data.

Even though a monolithic application goes through a conversion process towards a modern alternative, it is not enough to guarantee the fact that its data cannot be subject to a security breach. The stakeholders are currently challenged by the transition to the modern and interactive way of creating software, which results into the necessity of detaching from monolithic applications.

Generating an upward trend in nowadays development techniques, microservices architecture cannot represent a complete solution for public system applications if not used in a secure environment. In fact, none of the existing solutions in the market could be assessed as completed without a well-designed and well-implemented security layer. Even if microservices usage represents a huge step into the modern way of developing applications based on loosely coupled principles, it reveals a new perspective of how to correctly develop a public system application in order to reach the best possible version that will impress customers through user experience, flexibility, autonomy, usage benefits and security.

Implementing a security layer based on existing standard encryption principles may be considered a safe approach in present. If the most powerful encryption algorithms are taken into consideration and a cost and performance analysis is made, the stakeholders can successfully obtain a properly secured solution for public system applications.

Nevertheless, for a public system environment to be certainly protected against different exploits, the developers should foresee that a new era in information technology is about to change everyone’s perspective of how to integrate security into the application development stage. To go deeper into the subject, quantum computing represents a real threat regarding the way cryptography is implemented nowadays.

Moreover, it is scientifically proved that quantum computing has amazing processing capabilities, imposing itself as a qualified driver to generate disruptive changes in the perspective of standard encryption [3].

2. Public Key Infrastructure Concerns on Software Architectures

A clear understanding of the monolithic applications is required to comprehend the current necessity of transitioning to a different software architecture.

Dmitry Namiot and Manfred Sneps-Sneppe [4] underlined in their research the reduced productivity rate due to the large code base that will also reflect on the quality of the code, the very tedious continuous integration and deployment process since new build images have to be created every time, the difficulty in changing the stack of technologies in time and the strong dependency on the initial developing team, since new members are hard to accommodate.

Based on the representation exposed in Figure 1, the distinguishing feature of a monolith architecture is the inclusive characteristic of the end product to incorporate everything regarding application development. Business logic, application logic, technical implementations, security approaches are all linked together in a single monolithic structure.

![Figure 1. Monolithic application architecture](image)

This characteristic leads to different negative consequences such as high complexity, increased size, unexpected behaviours triggered by some small changes in totally different parts of the application due to the strong dependence between components or unexpected denials of service in case of isolated failures in some parts of the application. Also, a security threat may affect the whole application because of its limited or non-existent possibility to decouple functionalities.
Aiming to reduce the impact of these drawbacks, a viable alternative would involve migration to the microservices architecture. Anyway, although microservices offer effective solutions for most of the aforementioned issues just by its implementation philosophy, it is worth mentioning that usually the migration process tends to be more difficult when dealing with microservices. In addition, monolithic applications decomposition may cause unexpected, hard to solve problems due to improper implementation when different components do not allow separation of concerns and solution redesign may be the only alternative when as-is migration cannot be possible.

However, balancing risks and benefits, the result represents a major step forward into the realm of effective software development and for the purpose of obtaining a modern, robust and extensible application, known disadvantages of microservices architecture should be studied and addressed.

The most important disadvantages of a microservices architecture are the overall increase in complexity caused by developing independent instances which need to be well-defined and well-documented for future integrations, the necessity for more resources and a proper organizational culture that embraces the paradigm. In the particular case of public system applications, the institutional actors need to take ownership and professionally tackle each disadvantage.

Besides, microservices usage requires better understanding of security concerns, considering that insecure microservices represent an easy way to generate multiple security breaches that could jeopardize confidential data.

Striving for increased security would generally lead to a point where a decision has to be made – knowing the limitations and vulnerabilities of most encryption algorithms these days, based on what resources are needed and what kind of data has to be protected, developers have to trade security for performance. But what if suddenly the computational capabilities will drastically improve, and yet feasible protection measures will become obsolete as time required for breaking an encryption algorithm will significantly get reduced?

Researchers discovered decades ago that quantum computing will be able to solve two intractable problems that represent the building blocks of most cryptographic algorithms – large prime numbers factorization and discrete logarithm problems [3].

Moreover, a Cambridge University study has noticed that quantum computing would exponentially exceed today’s most powerful supercomputers in resolving quantum Fourier transformations [10]. Therefore, as researchers from the Air Force Institute of Technology concluded “computer scientists, physicists, engineers and experts from the industry are creating a growing community with the purpose of solving challenges defined by quantum computing. Their researches represent the beginning of the transition to post-quantum cryptography” [11].

3. Proposed Solution and Discussions

Perceiving the ideas presented above as identified issues in public system applications, this paper aims to provide a practical solution to assure citizens’ private data are protected, while applications are adjusted to modern development techniques that guarantee customer satisfaction via functional updates and intuitive user experience.

Firstly, it is necessary to bring applications to the newest development approaches so the most critical step of updating public system digital services consists of the actual migration, because a large number of these applications are defined by the ‘all in one’ principle. In addition, the older these are, the more complex these become [12], [13].

Secondly, the migration process may involve some different perspectives on how an application is developed and maintained. As an above all convention, a microservices architecture should guarantee Continuous Integration - Continuous Delivery to assure a fruitful outcome to all the parties concerned. Relying on this assumption, updated applications should respect a testing plan, maintain integrity of old systems, and offer the possibility to deliver new releases frequently.

Having this assumption in mind, cutting a monolithic application into microservices represents a long-term, difficult, but worthwhile process. By having application logic loosely coupled, developers work more efficiently on specific business units and get confidence that a minor change in the code will not have a major unknown impact on other components.

Microservices are perfectly matched with Service Oriented Architecture software design and represent a modern approach applied on client-server distributed programming.

“MSA arises from the broader area of Service Oriented Architecture (SOA) and focuses on specific
aspects, such as componentization of small services, application of agile practices for development, deployment and testing of services, usage of infrastructure automation with continuous delivery features, decentralized data management and decentralized governance among services.” [14].

Microservices architecture is suitable for public institutions in order to assure components availability and single responsibility principle as shown in Figure 2. It means that if a microservice which exposes some specific data about people does not work, it would not affect the entire application and public institutions can continue their activity, excepting that only one component needs to be addressed. This translates into higher availability times and reduced mean time to recovery, hence the incident is isolated.

![Figure 2. Microservices-based application architecture](image)

In addition, having public system applications designed based on microservices architecture allow public institutions to share data between them more efficiently. It means that two entities which serve the same institution can simultaneously access desired information. Also, it supports real time access to data and assures the integrity and authenticity of information.

In order to assure desired architectural functionalities, multiple development tools are based on this technical approach to develop microservices which offers the possibility to create small, single-scoped, independent components for public system applications. As an example, in a Java based development environment, Spring Boot enhances an effortless implementation, leverages the capability of configuring multiple components externally and settles down microservices monitoring, which can represent a good choice when developing microservices for the public sector.

The trend towards microservices has been supported by the increased number of devices that get interconnected through the Internet. Internet of Things has evolved from a niche concept to worldwide phenomenon, the number of devices growing exponentially, grace to cloud computing popularity.

Smart appliances, autonomous ground vehicles, sensor/beacon grids, VR devices, unnamed aerial vehicles represent quite novel entities that in a short time frame ended up being part of a technological ecosystem.

With so many devices connected to the Internet, the risk of cybercrimes skyrocketed, since ill-intended individuals try to exploit vulnerable targets to gain both prestige and financial benefits. Besides the direct consequences of a compromised infrastructure, there are several indirect aspects that are worth to be considered.

For example, a poorly protected network of beacons that is part of a public service responsible for providing indoor navigation/localization in a large venue, which exchanges information about specific user profiles through TCP/IP in an unprotected manner is a major risk for the users. A man in the middle would be capable of micro-targeting users, gathering data about their habits, about the time intervals when a subject uses the service, purposes for which the service is used, along with related information about the user profile. And while most people are aware about threats like phishing emails, malware attachments or running executable files from untrusted sources, the chances that someone would suspect a public service drop significantly. In addition, recent past has proved that some small breaches in public APIs could lead to massive data harvesting campaigns, user profiling and surveillance – Cambridge Analytica is being just an example.

So, in addition to specific development tools, production software experiences prove the fact that there are some ideas which also need to be taken into consideration when developing microservices and one of them is the need to integrate a microservices security layer. Some relevant key-characteristics of a security architecture are out of the box provided by the frameworks like Spring Security, Java EE or Platform Security. These bring a selection of features like secure communication protocols, annotations which establish security references, authentication, and authorization, all necessary for a properly designed project in terms of security. Some frameworks like Spring Security have an additional advantage represented by the fact that there is no need for containers to acquire security because it is granted by design.

All these aforementioned characteristics combined with some cryptography algorithms create a secure solution for nowadays public system application microservices. But taking into consideration future technological perspectives regarding quantum computer impact on security, microservices need an additional protection layer to deal with upcoming threats.
“It has been proved that lattice cryptography can be defined by properties such as high security guarantee, efficiency, resistance against future threats such as quantum computing and flexibility regarding homomorphic encryption-based tools” [15].

Lattice cryptography characteristics are defined around lattice problems which have an initial point represented by a multidimensional lattice.

A multidimensional space of a lattice problem can be defined as: “1. an additive subgroup: $0 \in L$, and $-x, x + y \in L$ for every $x, y \in L$; and 2. discrete: every $x \in L$ has a neighbourhood in $\mathbb{R}^n$ in which $x$ is the only lattice point.” [16].

The main principle is that one has a starting point in the multidimensional structure, and one ends up in a location based on spatial coordinates, but it is difficult to mention what other points are represented around someone when he or she is in that space. In order to create a multidimensional space, lattice problems define an offset and a multiplier for position vectors, a principle which gives immunity to lattice cryptography in front of a quantum computing attack.

“Examples of lattice problems like the closest vector problem (CVP) and the shortest vector problem (SVP) provide fascinating capabilities in terms of computational complexity”, as Oded Regev presents in his study for Princeton Institute for Advanced Study [17].

Lattice cryptography can be reflected in the idea of having a public and private key. The first one is represented by the starting point and the second one is represented by the final point. If a comparison in terms of performance is made, the usage percent of resources is close to that of other encryption algorithms, as studies confirm.

Lattice problems have been studied through different perspectives in years and distinct approaches are available and discussed to implement the best cryptographic solution.

With not even an attack known, NTRU Encrypt algorithm is supposed to be the safest choice and its implementation started from the short vector problem. “NTRU cryptosystem in terms of security is based on a principle which defines the possibility to search in a high dimension lattice a short vector. This problem has a high level of difficulty even for quantum computing and this is the reason why NTRU cryptosystem becomes a subject of interest for industry as an alternative to define the new era of cryptography.” [18].

Above-mentioned details regarding lattice problems may represent a starting point for public institutions to assure a secure environment even in the future, by offering Figure 1 cryptographic implementation which quantum computing cannot solve.

In this case, current implementations of security layers on microservices architecture may be improved with NTRU cryptosystem to provide increased security over forthcoming challenges. A practical solution can be implemented by creating a gateway between microservices, which controls the interaction. At this point, NTRU cryptosystem may be added to ensure an enlarged level of protection by creating a functionality based on tokens, which can be encrypted with NTRU encryption in gateway and sent in microservices to be decrypted. As an example, by using OAuth2, a consumer application calls the gateway with a generated token. The gateway receives the token on request, encrypts it using NTRU encryption and calls desired microservices with the encrypted request. These microservices receive the request, decrypt the token and check its validity. If everything is alright, then these microservices can securely retrieve desired information.

4. Conclusion

This paper aims to outline some ideas which can be implemented by public institutions with the purpose of offering an enhanced security in quantum computing context. Being aware of how fast technology progresses and how different the future will be in terms of development and security, the stakeholders need to upgrade their applications and offer the most reliable solutions for the users. As many studies have shown, there is an upward trend for implementing microservices and the architectural design fits on public system applications requirements, since independent components which can be shared between applications are provided. But adopting a modern way of development is not the only thing developers should be aware of. Understanding that quantum computers represent a real security threat, public institutions can obviate its consequences by timely implementing new security algorithms which are proved to be resistant in front of quantum computing. Studying, understanding, and implementing lattice-based cryptography will lead to a secure and efficient management of public system IT solutions in the context of an ongoing conflict with an impending future cryptographic threat.
References


