Scalable Front-End Architecture: Building for Growth and Sustainability

Oleksandr Tkachenko[1*], Aleksei Chechet[2], Maksim Chernykh[3], Sergei Bunas[4], Przemyslaw Jatkiewicz[5]
[1]Playtech, Sofia 1784, Bulgaria
[2]New Edge DWC-LLC, Dubai 00000, United Arab Emirates
[3]Boom Pay, Inc, Austin 78704, United States of America
[4]Supersales Global Pte Ltd, Singapore 049422, Singapore
[5]Department of Business Informatics, University of Gdansk, Gdansk 80-309, Poland
E-mail: otkachenko893@gmail.com[1], al.chechet@outlook.com[2], chernykh_maks@hotmail.com[3], sergeibunas@outlook.com[4], p.jatkiewicz@hotmail.com[5]
*Corresponding author

Keywords: Information security, structure with adaptive extensibility, interface design, platform development and reliability, ensuring privacy of customer data.

Received: May 30, 2024

The aim of the research is to develop strategies for building scalable interface architectures that ensure stability and protection of user data. The research identified the key characteristics of scalable interface architectures, such as ReactJS, GraphQL, and SSR. Within the scope of the research, principles for constructing scalable front-end architectures have been developed and described. These principles, such as modularity, componentisation, microservices architecture, adaptive and responsive design, as well as load management and scalability, data security, and fault tolerance, aim to ensure flexibility, efficiency, and user convenience across various devices and under different conditions. These principles enable the creation of architectures capable of adapting to changing requirements and scaling to ensure high performance and reliability when dealing with various types of data and workloads. The practical significance of the study is determined by the creation of a pragmatic guide for developers and architects of front-end systems who seek to create scalable applications.

Povzetek: Raziskava obravnava strategije za razvoj skalabilnih vmesniških arhitektur, ki zagotavljajo stabilnost in varnost uporabniških podatkov. Identificirane so ključne značilnosti, kot so modularnost, komponentizacija, mikroservisna arhitektura ter prilagodljiv in odziven dizajn. Ti principi omogočajo prilagodljivost, učinkovitost in uporabniško prijaznost na različnih napravah in v različnih pogojih.

1 Introduction

The examination of scalable interface architecture and data security is particularly pertinent in today's digital landscape. As the number of users on online platforms continues to surge and the volume of personal data proliferates, the imperative for secure systems becomes increasingly critical. The persistent advancement of Internet technologies, coupled with the evolving nature of cybersecurity threats, compels system developers to continuously refine their strategies for scalability and security. This ongoing evolution necessitates the development of innovative solutions that can effectively manage growing workloads while ensuring robust protection of user information. In this context, the ability to adapt to changing requirements and loads, maintain high performance, and safeguard data integrity and confidentiality are paramount. Consequently, the integration of advanced technologies and comprehensive security measures in scalable interface architectures is not only a necessity but a fundamental aspect of modern system development.

Thus, the study on the mentioned topic is crucial for effective adaptation to changing conditions and ensuring long-term success in the virtual environment. Furthermore, as the volume of data and the number of users on platforms increases, there are challenges in ensuring high performance, adaptability to change, and data protection. Improved approaches to front-end system design are needed to scale efficiently and ensure the security of customer data. The challenges range from online business to personal data privacy and require careful research to create innovative solutions.

Many other researchers addressed aspects related to this topic. The study by D. Dinev [1] highlights the rapid development and implementation of IoT devices, emphasizing the security and protection of user data. A promising technology in this field is Long Range (LoRa), which allows secure data transmission over long distances and offers extended battery life for IoT devices. M. Petkova [2] addressed various options for creating test platforms for innovative research, focusing mainly on the area of next-generation internet development. The study results in aim to provide generalised information about the structure and performance of such platforms.

W. Dimitrov and T. Ostrovska [3] investigated the use of Virtual Reality (VR) and Augmented Reality (AR) technologies in the context of creating scalable front-end architectures, with a focus on their growth and sustainability. The study examines various aspects of VR and AR application development, including software and hardware, developer tools, and user data protection issues. A. Urilski et al. [4] addressed some of the most popular and widely used e-learning systems. The authors emphasised the importance of reliability and protection of user privacy. D. Orozova [5] addressed new challenges of analysing and managing large amounts of data in distributed storage, as well as ensuring security, resilience, and privacy of information. The paper analyses data processing and analysis using the Map/Reduce paradigm. A. Tsenov [6] analysed architecture management in modern telecommunication and information systems, starting with the introduction and consideration of the basic principles of management in telecommunications. Models and methods for assessing the quality of service are presented as deliverables, including a hierarchical user experience model and a fuzzy logic method for multi-criteria evaluation.

As such, many researchers addressed various aspects in the area of data security and privacy, including the risks of data breaches and how to improve protection. However, these studies did not address issues related to the application of scalable architectures in front-end development security.

This article contributes to the field by providing a comprehensive analysis of scalable front-end architectures and data security in the context of modern digital environments. It not only evaluates various established approaches but also introduces practical insights into the implementation of these architectures in leading global companies. By addressing the challenges of scalability, stability, and data protection, the study offers valuable recommendations for developing resilient and secure interface architectures. This research seeks to close these gaps and propose innovative approaches for securing user data in scalable interface architectures. The study aims to develop methods for constructing scalable front-end architecture capable of ensuring stability and guaranteeing the protection of user data in the digital environment.

2 Materials and methods

The analysis method was used to examine existing scalable interface architectures, their features, advantages, and disadvantages. This method was used to conduct a systematic analysis of the literature and practical examples of the use of different architectural approaches. The analysis highlighted the main characteristics of each architecture and examples of their application in real projects. Moreover, this approach was used to explore data security issues and features of interface architectures, applying a wide range of technologies such as the IoT, test platforms, VR and AR, e-learning systems, big data analytics and telecommunication systems. Furthermore, aspects of smart buildings, optimisation of instruction caching, cloud development, application of deep neural networks, strategies for protecting sensitive information, semantic web, artificial intelligence, orchestration, and communication techniques, microservice architecture, native compilation platform for scalable architectures and distributed registry technologies were also addressed.

The experimentation method was utilised to develop and provide detailed descriptions of the principles underlying scalable front-end architectures. It involved creating conceptual models and conducting an iterative process of prototyping and validation. This experimental approach allowed not only to assess the effectiveness and applicability of each principle but also to identify their impact on the overall architecture and user experience. The results obtained served as the basis for formulating recommendations and establishing best practices in the field of scalable interface architecture development. Some of the illustrations were generated using the Matlab tool. Furthermore, a structural diagram was employed to visually represent the key components and relationships in various interface architectures. It encompassed aspects such as client-side and server-side components, data protection mechanisms, data management methods, as well as scalability capabilities and system stability assurance.

The comparison table was a comprehensive analytical process that compared key characteristics of different scalable interface architectures, such as those used by Netflix, Airbnb, Facebook, Amazon, and Google. Key aspects of each of the architectures were covered to create the table, including their structure, data processing methods, scaling mechanisms, technology utilisation and more. Each of the architectures was thoroughly analysed, highlighting features, advantages and disadvantages. As such, a detailed comparison between the different architectures was created and their key differences were identified. This approach provided a complete understanding of how each architecture functions and what their peculiarities are. This level of detail is necessary for selecting the best architecture depending on the specific requirements and goals of the project. Thus, this method facilitated mastering the concept of scalable interface architectures, their implementation, analysis of real data and comparison of different approaches, which allowed to highlight recommendations and draw certain conclusions.

3 Results

3.1 Introduction to scalable interface architecture

In the contemporary digital landscape, the proliferation of online platforms and the exponential growth of user data demand efficient management of data processing and storage. Ensuring high performance and security of transmitted information is crucial, particularly as systems must remain adaptable to evolving requirements and fluctuating loads. Scalability and stability of interface architecture are paramount in delivering a high-quality user experience. Scalability ensures that systems can handle increasing amounts of work by adding resources, thus maintaining performance levels. Stability ensures that systems operate reliably under various conditions, preventing failures and minimizing downtime. Together, these aspects ensure that users experience smooth and uninterrupted service, regardless of the volume of data or the number of users accessing the platform [7].

Table 1: Explanation of key terms in scalable interface architectures

Term	Explanation
Scalability	The ability of a system or application to efficiently increase its resources and bandwidth to handle growing workloads.
Interface Architecture	The structure and organization of the front-end components of an application, which affects its behavior and user interaction.
Stability	The ability of a system to maintain normal operation and prevent failures even under changing conditions and loads.
Data Protection	Measures and techniques to ensure the confidentiality, integrity, and availability of user information from unauthorized access or damage.
Component-Based Architecture	An approach to software development where the interface is composed of independent components that can be reused in different parts of the application.
Application Programming Interface (API)	A set of tools and protocols for the interaction of different applications, enabling integration and interaction of system components.
Caching	A mechanism for storing data to speed up access to it in the future, which enhances the performance of the system.

There are certain prominent terms in the context of this topic. Scalability is the ability of a system or application to efficiently increase its resources and bandwidth to handle growing workloads. Interface architecture defines the structure and organisation of the front-end components of an application, which affects its behaviour and interaction with the user. Stability refers to the ability of a system or application to maintain normal operation and prevent failures even under changing conditions and loads. Data protection includes measures and techniques to ensure confidentiality, integrity and availability of user information from unauthorised access or damage. Component-based architecture involves developing software from independent components, which facilitates scaling and modification of the application [8]. API is a set of tools and protocols for the interaction of different applications. The modularity of the system enables splitting it into independent components, which simplifies development and support. Caching is a mechanism for storing data to speed up access to it. Design patterns are repeatable solutions to typical problems in software design that help improve its quality and simplify development [9, 10].

Several approaches for scalable front-end architectures and data security should be considered. A micro-frontend architecture involves dividing the interface into separate, independent modules, enhancing scalability and allowing for the integration of new features without disrupting the overall application. Single-page applications (SPAs) load all necessary HTML, and JavaScript at once, improving performance and user experience while utilizing modern authentication and authorization methods for data security. Using a Content Delivery Network (CDN) reduces resource load times and enhances scalability, providing protection against DDoS attacks and data encryption [11]. Server-side rendering (SSR) processes HTML on the server, improving Search Engine Optimisation (SEO). and reducing initial page load time, thereby better controlling data security. Containers and orchestration tools like Docker and Kubernetes facilitate easy scaling and resilience of applications by isolating containers to improve security. The choice of approach depends on the specific project requirements and the needs for scalability and data security [12].

Nevertheless, existing approaches to developing interface architectures are not always able to provide an optimal combination of scalability, stability, and data security. Difficulties arise due to the variety of platforms on which web applications are deployed and the increasing volume and variety of data that needs to be processed and transmitted. In light of these challenges, it becomes necessary to develop new approaches to build an interface architecture that can adapt to the growing requirements and ensure the security of user data. It is also necessary to consider that effective scalability and data security requires an integrated approach that includes not only the selection of appropriate technologies and methods but also the right architectural concept and constant attention to system upgrades and enhancements.

3.2 Principles of building scalable interface architectures

Scalable interface architectures encompass various principles of construction. For example, modular interface architecture involves an approach where the user interface is divided into separate modules, each performing a specific function or task (Figure 1). These modules are independent of each other and can be easily replaced, modified, or added without the need to rewrite the entire interface. Each component in modular architecture has clearly defined functional boundaries, making them easier to understand and support. The use of modular architecture provides flexibility and scalability to the system, allowing the interface to adapt to changing user and business needs. Additionally, modular architecture promotes code reuse, speeding up the development process and ensuring uniformity and consistency in the user interface.

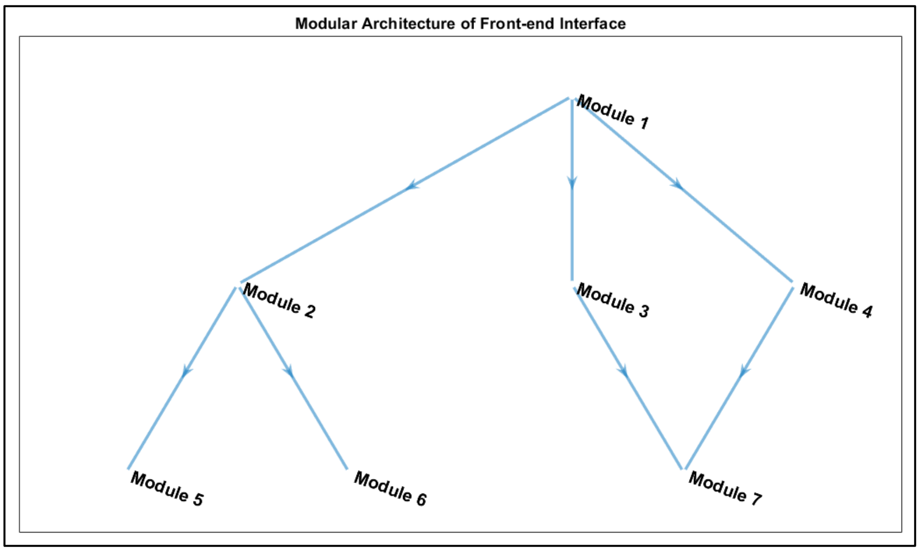

Figure 1: Modular interface architecture

This graph represents a modular architecture of the front-end interface, where each module performs a specific function or task, and the connections between modules depict the interaction between them. Modular architecture allows for easy scalability of the system and reuse of interface components to create more complex user interfaces.

Component-based architecture of the interface is an approach where the user interface is composed of a set of ready-made components such as buttons, text fields, lists, and other elements [13]. These components are independent and reusable building blocks that can be easily combined and arranged to create various interfaces. Components provide clearly defined functionality and well-designed API’s, making them easy to use and integrate into projects. This approach simplifies interface development because developers can focus on creating and configuring components without spending time implementing each interface element from scratch. Additionally, component-based architecture promotes code reuse and ensures consistency in the appearance and behavior of the interface, facilitating the creation of cohesive and easily maintainable applications [14]. Indeed, component-based architecture of the interface provides modularity, reusability, flexibility, and customisation of the interface, making it an effective approach to user interface development.

In turn, microservices architecture is an approach to building a user interface where functionality is divided into separate independent components called microservices [15]. Each microservice is responsible for executing a specific part of the interface’s functionality, allowing developers to create more flexible, scalable, and easily maintainable systems. Microservices can be developed, deployed, and scaled independently of each other, providing a high degree of flexibility and scalability to the system. This approach helps to improve the system’s fault tolerance, as failures in one microservice do not significantly impact the operation of other parts of the interface. Moreover, microservices architecture promotes code reusability and enhances the development process, making it more modular and easier to understand [16-18].

Adaptive design is an approach to user interface creation that automatically responds and adapts to various usage conditions, including screen resolution and device type. This principle ensures convenience and efficiency in using the interface across different devices, such as smartphones, tablets, and desktop computers. Flexibility and adaptability in systems are achieved through the use of scalable and customisable components, as well as mechanisms for automatic adjustment and scaling. This approach enables optimal user interaction with the interface regardless of their device and usage context, thus increasing user satisfaction and overall system efficiency.

Equally important is responsive design, which is a methodology for user interface development that dynamically responds to changes in the size and resolution of the user’s device screen (Figure 2).

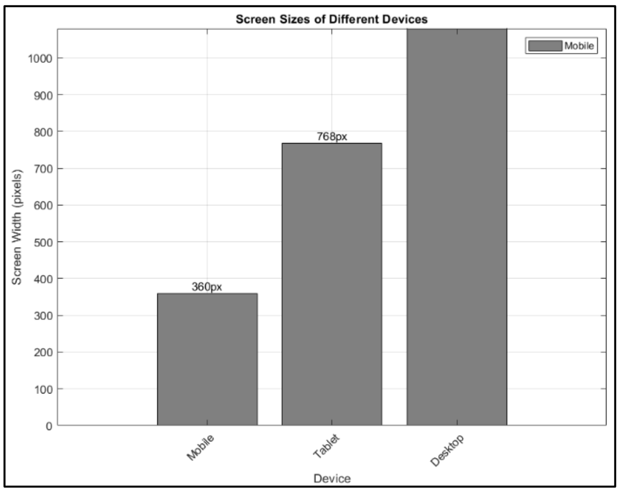

Figure 2: Responsive design

Horizontal and vertical scaling methods are essential approaches to increasing a system's capacity and performance. Horizontal scaling (scaling out) involves adding new servers or nodes to an existing system to increase computational power. This allows the system to handle more requests or data by distributing the load across multiple servers. Horizontal scaling is effective for large, distributed systems and can be implemented with minimal downtime. However, it requires managing distributed data and ensuring consistency across nodes. Vertical scaling (scaling up) involves increasing the resources of an existing server, such as adding more CPUs, memory, or storage. This enables a single server to handle larger workloads or perform more computations. Vertical scaling is simpler to implement since it doesn't require data distribution, but it has physical limitations and can become expensive at higher levels. Additionally, it introduces a single point of failure, as the system relies on one server. Scalable interface architectures involve designing systems to efficiently handle increasing loads and data volumes without sacrificing performance. Key principles include:

- Modularity: Dividing the user interface into independent modules that perform specific functions, allowing for easy scaling and modification without affecting the entire system.
- Component-based architecture: Building the interface from reusable components, streamlining development and ensuring consistency across the application.
- Microservices architecture: Splitting functionality into independent services that can be developed, deployed, and scaled separately, enhancing flexibility and resilience.
- Adaptive and responsive design: Ensuring the interface automatically adjusts to different devices and conditions, providing an optimal user experience across various platforms.

This principle ensures optimal content display and interface layout on any device, ranging from mobile devices and tablets to desktop computers and televisions. By using flexible grids, images, and media queries, responsive design enables the creation of universal interfaces that automatically adapt to different screens and provide users with comfortable and intuitive interaction with the application or website [19].

This bar chart illustrates the screen width of various devices – mobile phones, tablets, and desktop computers – in pixels. Each bar in the chart represents one of the devices, with its height corresponding to the screen width. Resilience and fault recovery are key aspects in designing scalable interface architectures. These principles aim to ensure continuous system operation even in the event of failures or disruptions. To achieve resilience, load distribution and resource redundancy should be implemented, enabling the system to maintain functionality and ensure service availability even under adverse conditions [20]. Fault recovery mechanisms include automatic service and data restoration, rapid problem diagnosis, and the utilisation of backup resources to minimise system downtime. These principles contribute to increasing the reliability and stability of interface architectures, ensuring uninterrupted operation even under extreme loads or failures [21].

Attention should also be paid to data security and access control, which is one of the fundamental principles in building scalable interface architectures (Figure 3). This principle aims to ensure the confidentiality and integrity of data, as well as control access to it. Data encryption methods, authentication, and authorisation mechanisms play a crucial role in preventing unauthorised access and use of information. Data encryption ensures its protection during transmission and storage, while authentication and authorisation mechanisms define user access rights to specific system resources and functions. This approach helps minimise the risks of information leakage and ensures compliance with modern data security standards. Effective implementation of the data security and access principle is an integral part of creating reliable and secure interface architectures.

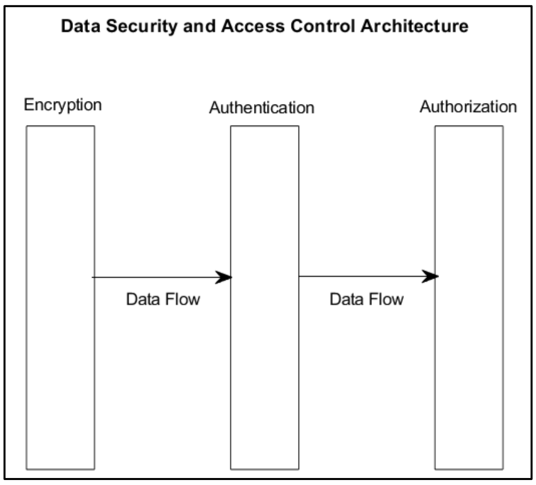

Figure 3: Data security and access control

This graphical representation of the security architecture illustrates key components (encryption, authentication, authorization) and the flow of data between these components. Such visualisation helps understand the structure and interaction of key elements in the data security and access control system.

Load management and scalability are crucial aspects to consider when designing scalable interface architectures (Figure 4). They ensure the system’s ability to efficiently handle increasing volumes of requests and data. Horizontal and vertical scaling methods allow increasing system performance by adding computational resources or distributing the workload among different components. Resource and request management help optimize the utilization of hardware and software resources, maintaining a balance between system performance and efficiency. As a result, the system can maintain high responsiveness and availability even under significant loads, which is a key requirement for modern applications and services.

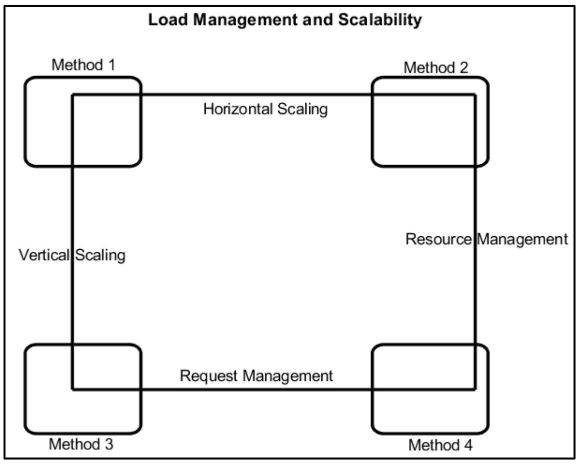

Figure 4: Load management and scalability

This visualisation illustrates various management methods and shows the relationship between them. Each block is accompanied by a text label explaining its purpose. It helps understand key aspects of load management and scalability, such as horizontal and vertical scaling, resource and request management. To visualise the fundamental components and principles underlying scalable architecture, it is also worthwhile to consider a general structural diagram of web applications (Figure 5).

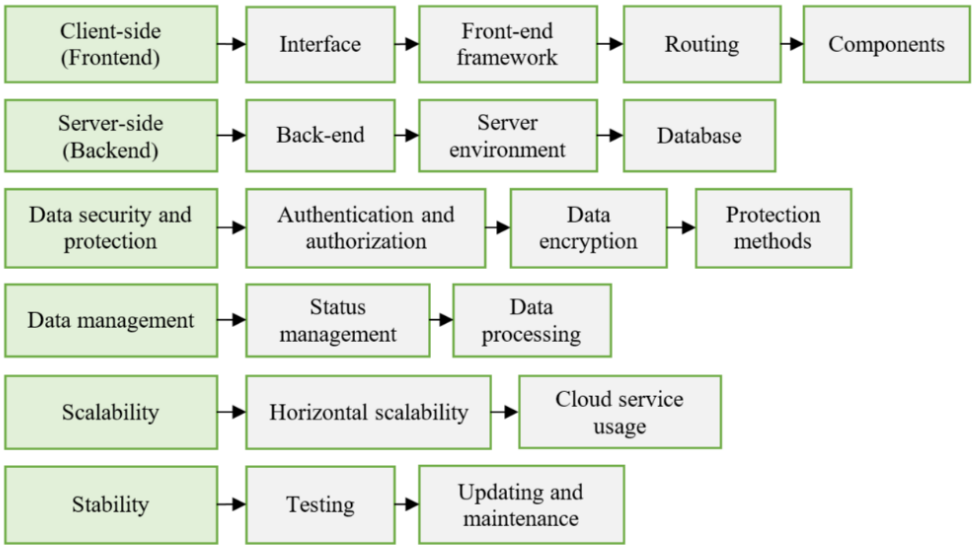

Figure 5: General architecture diagram of a scalable web application

Such a diagram helps better understand the overall architecture of the application, identify key components, and define the main data flows in the system.

3.3 Real data analysis and comparison of approaches

Scalable interface architectures are an important aspect of frontend development, enabling the creation of flexible and extensible applications. Several types of such architectures can be distinguished. For example, a modular architecture divides an application into independent modules, each of which is responsible for a specific functionality. This allows developers to work on different modules in parallel and easily scale the application if necessary. Component-based architecture divides the interface into independent components that can be reused in different parts of the application. This approach helps create flexible and scalable interfaces where each component is responsible for its functionality. In turn, microservice architecture divides the application into small, independent services, each of which performs a specific function.

Thus, this architecture flexibly scales and deploys individual parts of the application and provides easy system support and extension. A pure architecture offers the division of the application into layers, each with its responsibilities and dependencies directed inward, allowing the creation of highly connected but loosely coupled components, and facilitating the testing, maintenance, and scaling of the application. In addition, microservices architecture divides an application into small, self-contained services, each of which performs a specific function and can be deployed and scaled independently. This enables flexible and scalable systems where each service can be developed, deployed, and scaled independently. Model-View-Presenter architecture divides the application into a model, a view, and a presenter. This architecture separates the application logic from its presentation, making it easier to test and maintain the application. The Redux architecture proposes the use of unidirectional data flow to manage the application state, allowing for predictable and scalable interfaces where the application state is stored in a single repository and is only changed through actions [22].

Furthermore, a separation of duties architecture should also be added. The principle of segregation of duties involves breaking each application layer into smaller sections, each of which solves only one application problem or function. Scalable logging also exists. When moving from heavy monoliths to distributed computing systems, log monitoring becomes difficult [23]. A distributed approach helps developers to combat problems and also log as much information as required. Equally important is the principle of sole responsibility. This principle suggests that each class should have only one reason for the change, which promotes clean, reproducible code with fewer bugs. These are just some examples of scalable interface architectures and principles. Depending on the requirements and specifics of the project, it is possible to select the appropriate architecture or a combination of several architectures to achieve the desired level of scalability and flexibility.

Aside from examples of architectures and principles, it is worth considering real-world examples of scalable interface architectures (Table 2). For instance, Netflix uses a microservice architecture with the front end divided into many independent modules. Netfilx uses ReactJS to create dynamic interfaces and GraphQL for API management and data caching. Netflix’s architecture scales horizontally by adding new servers to the pool. Airbnb, on the other hand, uses a SSR based architecture with ReactJS to optimise page load time and SEO. They also use GraphQL for API management and data caching and scale horizontally by adding new servers to the pool. Facebook, on the other hand, uses ReactJS to create dynamic interfaces and GraphQL for API management and data caching. They have also developed their content management system called GraphQL Relay. Facebook's architecture scales horizontally by adding new servers to the pool. Amazon also uses a microservice architecture to split the front end into independent modules. They use ReactJS and AngularJS to create dynamic interfaces and use Amazon Web Services (AWS) Lambda for server-side functions. Amazon's architecture also scales horizontally by adding new servers to the pool. Lastly, Google uses ReactJS and AngularJS to create dynamic interfaces and uses Google Web Toolkit (GWT) to compile Java code into JavaScript. They also use Google Cloud Platform (GCP) for hosting and scaling. Google's architecture scales horizontally by adding new servers to the pool.

Table 2: Comparative table of scalable front-end architectures

Company	Peculiarities	Pros	Cons
Netflix	Microservice architecture. Use of ReactJS and GraphQL. Horizontal scalability	High flexibility and scalability. Effective data management using GraphQL.	High infrastructure configuration requirements Specialised skill-set requirements
Airbnb	SSR and ReactJS. GraphQL for API management. Horizontal scalability	Optimized page loading and SEO Flexibility in data management with GraphQL	Additional server infrastructure maintenance expenses
Facebook	Use of ReactJS and GraphQL. "GraphQL Relay" content management system. Horizontal scalability	High performance and flexibility. Effective data management via GraphQL	Requires substantial resources for development and upkeep
Amazon	Microservice architecture. ReactJS and AngularJS. Horizontal scalability via AWS Lambda	High flexibility and scalability. Resource-saving using AWS Lambda	High AWS configuration and infrastructure management requirements
Google	ReactJS, AngularJS, and GWT Horizontal scalability via GCP	Variety of technology and tools. High performance using GCP	Difficulty in deploying and maintaining applications on GCP

This table can help organise information and make a more informative comparison between different scalable interface architectures. Furthermore, it is possible to highlight the basic principles of scalable interface architectures. It is worth considering the use of microservice architecture, which involves splitting the front end into independent modules. As well as the use of ReactJS, a widely used JavaScript library, for developing dynamic interfaces. The implementation of GraphQL as a query language for efficient API management and data caching should be emphasised. An important aspect is the use of SSR to optimise page load times and improve SEO performance. And horizontal scaling, which involves adding new servers to the pool to increase performance.

3.4 Recommendations

After analysing various scalable interface architectures, several recommendations for their further improvement can be identified. To improve the reliability and stability of the architecture, extensive use of automated tests and Continuous Integration/Continuous Delivery (CI/CD) tools is recommended. This will help identify and fix problems early in the development process. It is important to continue to research and develop methods to optimise interface performance. This includes optimising page loading, reducing response times and improving the overall responsiveness of the user interface. To meet the growing needs of developers and users, it is recommended that the functionality of the architecture be continually enhanced. This may include adding support for new technologies, expanding APIs, and improving development tools. In addition, it is important to ensure the availability of tutorials and documentation for developers, as well as to provide quality technical support. This approach will help accelerate the process of implementing and improving the architecture, as well as improve the skills of employees.

4 Discussion

For a thorough review of this topic, other studies on scalable front-end architecture should be consulted. As such, L. Chamari et al. [7] considered modern smart buildings that require integration of various information systems such as Building Management Systems, Energy Management Systems, IoT, Building Information Models and others for effective management and monitoring. However, the lack of a well-defined system architecture with APIs poses challenges for developing and deploying effective applications for smart buildings. The architecture proposed in this paper, based on the principles of microservices, APIs, cloud components and full-stack partitioning, aims to enhance the repeatability, modularity, and scalability of such applications. The research results include the development of three smart building applications that demonstrate the reusable and modular aspects of the proposed architecture. That is, the common aspects between the referenced work and this study lie in the endeavour to develop scalable applications using modern architectural principles. However, the difference is that this study emphasises ensuring the stability and security of user data, which is a critical aspect in application development, especially in the context of the growth of Internet technologies and cybersecurity threats. The study Y. Degawa et al. [22] addresses the topic of scalable front-end architecture, focusing specifically on smart buildings and the integration of different information systems, and draws attention to the challenges of developing effective smart building applications due to the lack of a well-defined system architecture with APIs.

Y. Degawa et al. [22] emphasised that processor performance depends on the instruction delivery front end, as the design of interface architecture requires consideration of instruction processing performance and cache management in processors. Various techniques such as cache replacement algorithms and prefetching have been investigated to improve performance. One key factor is to reduce instruction cache misses. However, the number of misses does not always explain the performance changes due to parallel processing of misses by the front end. The study proposes a new factor, the number of missed areas, which explains the performance variation. Hence, both studies address the performance and scalability issues of the processor front end and recognise the importance of considering the performance of instruction processing and cache management to design a scalable architecture. However, unlike this study, which focuses on designing a scalable interface architecture to protect user data, the research concentrates on analysing the number of cache misses and introducing a new factor, the number of miss areas.

P. Mwiinga [24] pointed out that in today’s world, saturated with technological innovation and extensive digital interaction, preserving user privacy, and providing robust security measures are becoming increasingly important. The study analyses technologies aimed at preserving user privacy and examines their key role in balancing security and privacy protection. Through a detailed examination of various techniques and innovations in this area, the study assesses the challenges, achievements and prospects associated with achieving an optimal balance between security and user privacy in the digital space. Thus, both studies focus on the importance of user privacy and security in the context of the digital environment. They also approach this issue by analysing technologies aimed at preserving user privacy and evaluating their role in balancing security and privacy. However, while the 2023 study is more focused on analysing the challenges, advances and prospects in security and privacy, this study specifically aims to develop a scalable interface architecture with a focus on stability and protection of user data.

K. Khan [25] proposed a comprehensive guide to cloud development suitable for both experienced professionals and novices. He reviewed the key concepts of cloud applications and explored architectural patterns and their development techniques. The work also discusses DevOps practices and the importance of securing and monitoring applications in the cloud, real case studies and future trends in cloud development. Both studies address important aspects of software development using advanced technologies. While the aforementioned study focuses on cloud development and DevOps practices, this study focuses on scalable interface architecture with growth and resilience in mind and user data protection. However, both approaches complement each other, as cloud development and DevOps practices can be effectively integrated into a scalable UI Architecture, ensuring application stability and security.

R. Davis [26] emphasized the ubiquitously used deep neural networks which are capable of more accurate performance on various tasks compared to manual systems. He emphasised that this has increased the need for more sophisticated models to handle large amounts of data, but traditional computing architectures have faced a performance bottleneck due to the need to move data around. A new approach using an optical neural network with multiple analogue frequency conversions is proposed in this paper. The performance of the system has been successfully demonstrated on a three-layer Deep Neural Networks for Modified National Institute of Standards and Technology (MNIST) digit analysis, which confirms its scalability and the possibility of all-analogue signal processing from start to finish.

K. Lawal et al. [27] predicted that by 2030, the number of IoT devices connected to the Internet will exceed 125 billion units. This will create large amounts of data and require scalable architectures. The study demonstrates the advantages of fog computing over cloud computing in scalable scenarios. The fixed building architecture limits the benefits of fog computing due to the low number of IoT devices. However, in scalable scenarios, the benefits of fog computing are significant. Thus, both studies point to the need for scalable data processing solutions.

E. Blessing and H. Klaus [28] focused on the implementation of face-covering detection technologies in public places, which raises questions about data security and privacy. This work examines strategies to protect sensitive information and considers the technical and ethical aspects of using the technology. Security measures adopted include encryption and access control, while privacy measures aim to preserve user rights and autonomy. Legal compliance and educational initiatives also play an important role in ensuring the ethical use of technology [29]. In summary, both studies address data security and privacy issues in the context of emerging technologies. However, this paper focuses on the development of a scalable interface architecture, while the other focuses on the implementation of face-covering detection technology in public spaces.

D. Recupero et al. [30] indicated that recent breakthroughs in computer science, such as the semantic web and artificial intelligence, have enabled the development of a flexible human-like service architecture based on the Zora service. This architecture consists of three modules to interact with humans and perform various tasks. It is flexible and can be extended with new modules, and inbuilt modules can be easily augmented with learning resources. Each module has a frontend for interacting with people and a backend for performing computations on the server. That is, both studies emphasise the flexibility of their architectures and the ability to extend them with new modules. However, this study focuses on the security and scalability aspects of the interface architecture, while the 2019 study focuses more on user interaction and task execution.

Y. Perera and D. Jayasuriya [31] identified that the use of web applications has increased significantly in recent times, generating interest in improving their usability. Frontend performance plays an important role in this, as it is the interface for users. To improve frontend performance and scalability, micro-frontend architectures are used to enable developers to deal with various issues such as orchestration and communication between components. Advanced technologies such as ReactJS and NodeJS are used to implement such architectures. Recent research has proposed new orchestration and communication techniques, resulting in significant improvements in web application performance. So, both studies highlight the increasing usage of web applications and also emphasise the importance of frontend performance and scalability of systems. However, this study focuses on designing a scalable interface architecture with an emphasis on user data protection. While other work focuses on using micro-frontend architecture to improve frontend performance and scalability.

O. Nikulina and K. Khatsko [32] emphasise that the modern development of various web systems uses new microservice architecture to improve features such as performance and portability. This requires the conversion of legacy systems from monolithic architecture to microservice architecture, which is a complex and costly process. This research aims to develop a method to apply micro-frontends to improve monolithic single-page applications. The authors proposed a method to transform the architecture of a software system using microservices. The approach used is based on reverse engineering and moving to a modular architecture to improve the performance and portability of the application. The study resulted in a 200% performance improvement and stabilised communication time between application components. Thus, both studies emphasise the need to use modern approaches to improve the performance of web-based systems and suggest certain methods for transforming the architecture of software systems using new technologies. However, this study focuses on developing a scalable front-end architecture that provides stability and protection of user data, while the second study focuses on improving monolithic single-page applications using micro-frontends.

Similar to previous authors, A. Pant [33] confirmed that securing sensitive data has become a priority for everyone from individuals to businesses and governments. This paper discusses the critical importance of data security and privacy compliance in today’s society. As technology continues to evolve, changing the way information is created, collected, and used, threats to data security are becoming increasingly serious [34; 35]. The author emphasises the multifaceted importance of data security and privacy compliance in various fields of endeavour. The importance of protecting personal information to protect privacy rights and maintain public trust is discussed first. Both the frequency and complexity of cyber threats are increasing, highlighting the importance of data security measures to prevent unauthorised access and information leaks. The general aspect of the research is the importance of data security in today’s digital world. Furthermore, the increasing threats to data security in the context of technological advancement are also pointed out. However, this study focuses on the development of a scalable interface architecture that ensures stability and security of user data, while the 2023 study focuses on a general analysis of the critical importance of data security in today’s society. Thus, while both studies address the issue of data security, their methodologies and focuses are different.

N. Paraskevopoulos et al. [36] introduced SpinQ, the first native compilation platform for scalable spin-qubit architectures. SpinQ addresses the unique operational constraints of the crossbar, enables scalable compilation, and achieves O(n) computational complexity. The authors analysed the performance of SpinQ on this architecture by compiling a variety of quantum circuits, revealing unique insights into the matching and architecture. The proposed new mapping techniques may improve the execution success of algorithms on this architecture and inspire further research in this area. That is, both papers present innovative approaches to developing architectures for specific domains. While this study focuses on the design of a scalable interface architecture for securing user information, the other focuses on the development of a compilation platform for scalable spin-qubit architectures. In doing so, both studies highlight the importance of considering the unique constraints and features of the target architectures.

S. Naqvi and A. Mohsin [37] considered an IoT system that has vulnerabilities such as scalability, security, and access control due to centralised architecture. To solve these problems, Distributed Ledger Technology (DLT) is used to provide a secure and scalable infrastructure for IoT. This research proposes the use of DLT based on Directed Acyclic Graph (DAG) to overcome the limitations of IoT devices. The authors presented an architecture based on the Tangle DAG framework and utilising a lightweight Masked Authentication Message data model. Experiments confirmed the energy-efficient execution of computations on the entire node, which allows efficient resource utilisation. As a result, a scalable access control infrastructure for IoT using a DAG-based distributed registry can be proposed. It comes out that both studies address the issues of security and scalability in different information systems. In both cases, the use of advanced technologies such as DLT and microservice architectures play a key role in achieving these goals. The main similarities of both works lie in the pursuit of security and scalability in information systems. While this study focuses on frontend aspects, providing stability and protection of user data, the study focuses on scalability, security, and access control aspects in centralised architectures. That is, this study proposes a scalable architecture for the interface and the second study proposes a scalable access control infrastructure for IoT.

T. Krishnappa [38] pointed out that the rapid growth and widespread use of social media have created new security and privacy challenges for individual users. The study aims to assess users' awareness of security and privacy measures and their understanding of potential risks and vulnerabilities. The results obtained are analysed to identify trends, knowledge gaps and user perceptions to help better understand the situation in this area. The results of the study can be used to develop recommendations for improving security measures and encouraging responsible online behaviour among social media administrators, legislators and users. The main objective of this study is to provide users with the necessary knowledge to ensure their protection in the digital age, emphasising their awareness and understanding of security and privacy in social media. Both works address important aspects of the information technology field but from different perspectives. This study focuses on the implementation of an interface scalable architecture with an emphasis on user data protection and stability. Furthermore, practical solutions are proposed to create robust systems that can scale efficiently and ensure data security. On the other hand, the 2023 survey highlights the security and privacy issues in social media and the level of user awareness of these issues. The author proposes a qualitative survey to assess users' understanding of security and privacy measures and to identify trends and gaps in their knowledge.

The general trend, confirmed by analysing various studies, indicates that data security and privacy are important criteria in digital technologies. However, given the evolution of the information space, attention must also be paid to user awareness of security and privacy measures within scalable interface architectures. This means that not only the technical aspects of security are crucial, but also the user's awareness of their role in data protection. Such an approach will lead to more resilient and secure information systems, as well as to a greater overall level of digital literacy.

5 Conclusions

The research conducted examined various aspects of scalable interface architectures and their impact on data security and privacy. The main topic addressed in the study was the issue of data security and privacy in the context of new technologies. Study results confirmed that these aspects are becoming increasingly important in the digital age, requiring serious attention from developers, businesses, and society as a whole. It is important to note that users' understanding of data security and privacy measures is a key factor in the successful implementation of scalable interface architectures.

One of the important study results is the development of methods and recommendations for ensuring data security and privacy in scalable interface architectures. The practical significance of such recommendations lies in the possibility of creating reliable and secure systems that can efficiently scale and protect user data. In addition, the study not only raised issues of user data security, but the growth and resilience of architectures were also important aspects. Data security is a key element in modern information technology, but it is also important to ensure that architectures are resilient and scalable to ensure their long-term performance and success in different usage scenarios.

Summarising the results and conclusions of the study, it is possible to note that the development of methods for creating scalable interface architectures plays a key role in the creation of reliable, productive, and flexible web applications. The right choice of architecture and its continuous improvement are essential for the successful implementation of projects and their competitiveness in the market. Further research in this area will contribute to the development of new methods and approaches and help solve current problems in interface development.

References

[1] D. Zh. Dinev, "LoRaWAN technology – The necessary solution to the challenges of wireless IoT networks," *Comput. Sci. Technol,* vol. 19, no. 1, pp. 23-30, 2021.

[2] M. Petkova, "Approaches for creating infrastructures for network and Internet innovations," *Yearb. Telecommun*, vol. 6, pp. 147-160, 2019.

[3] W. Dimitrov and T. Ostrovska, "Overview of virtual and augmented reality application development tools," 2020. [Online]. Available: https://doi.org/10.13140/RG.2.2.29293.82407/1

[4] A. Urilski, A. Malinova, and A. Rahnev, "Security threats and protection in E-Learning systems," in *Ann. Int. Sci. Conf. "Comput. Technol. Appl.",* Pamporovo, Bulgaria, 2021, pp. 115-129.

[5] D. A. Orozova, "Challenges of big data and big data analytics," *Ann. Burgas Free Univ*. vol. 28, pp. 47-52, 2018.

[6] A. Tsenov, "Management architectures in modern telecommunication and information systems," 2021. [Online]. Available: http://telecom.tu-sofia.bg/pdfs/Monograph%20Alexander%20Tsenov.pdf

[7] L. Chamari, E. Petrova, and P. Pauwels, "An end-to-end implementation of a service-oriented architecture for data-driven smart buildings," *IEEE Access*, vol. 11, pp. 117261-117281, 2023.

[8] M.-N. Tran and Y. Kim, "Optimized resource usage with hybrid auto-scaling system for knative serverless edge computing," *Fut. Gener. Comput. Syst*, vol. 152, pp. 304-316, 2024.

[9] N. Dhieb, H. Ghazzai, H. Besbes, and Y. Massoud, "Scalable and secure architecture for distributed IoT systems," in *2020 IEEE Technol. Eng. Manage. Conf. (TEMSCON),* Novi, MI, USA, 2020, pp. 1-6.

[10] D. Sabol and J. Skalka, "The architecture of visual design in modern web applications," in E. Smyrnova-Trybulska, P. Kommers, M. Drlík, and J. Skalka,

Eds., *Microlearning*. Cham, Switzerland: Springer, 2022.

[11] A. Quraishi, M. A. Rusho, A. Prasad, I. Keshta, R. Rivera, and M. W. Bhatt, "Employing Deep Neural Networks for Real-Time Anomaly Detection and Mitigation in IoT-Based Smart Grid Cybersecurity Systems," in *3rd IEEE Int. Conf. Distr. Comput. Electr. Circ. Electron., ICDCECE 2024*, Hybrid, Ballari: Institute of Electrical and Electronics Engineers, 2024. Available: https://doi.org/10.1109/ICDCECE60827.2024.10548160

[12] K. Li, Y. Ding, D. Shen, Q. Li, and Z. Zhen, "The design and research of front-end framework for microservice environment," in *2020 Int. Conf. Comput. Inf. Big Data Appl. (CIBDA)*, Guiyang, China, 2020, pp. 124-127.

[13] I. Aviv, R. Gafni, S. Sherman, B. Aviv, A. Sterkin, and E. Bega, "Infrastructure from Code: The Next Generation of Cloud Lifecycle Automation," *IEEE Software*, vol. 40, no. 1, pp. 42-49, 2023. Available: https://doi.org/10.1109/MS.2022.3209958

[14] I. Aviv, R. Gafni, S. Sherman, B. Aviv, A. Sterkin, and E. Bega, "Cloud Infrastructure from Python Code – breaking the Barriers of Cloud Deployment," in *17th Eur. Conf. Software Archit., ECSA 2023*, Istanbul, Turkey; 2023. Available: https://conf.researchr.org/details/ecsa-2023/ecsa-2023-journal-first/2/Cloud-Infrastructure-from-Python-Code-breaking-the-Barriers-of-Cloud-Deployment

[15] O. Titova, P. Luzan, N. Sosnytska, S. Kulieshov, and O. Suprun, "Information and Communication Technology Tools for Enhancing Engineering Students' Creativity," in *Lect. Notes Mech. Engin*, Cham: Springer, 2021, pp. 332-340. Available: https://doi.org/10.1007/978-3-030-77719-7_33

[16] R. Frankiv, ""Perfect presence space": Theoretical and practical aspects of the concept," *Archit. Stud*, vol. 10, no. 1, pp. 17-23, 2024.

[17] Yu. Dyba and Yo. Rotchniak, "Evolution of the buildings layout of the Lviv Main Railway Station," *Archit. Stud*, vol. 10, no. 1, pp. 35-46, 2024.

[18] A. Kucher and V. Mazurenko, "Essence and features of economic security of the industry sector," *Dev. Manag*, vol. 23, no. 2, pp. 16-24, 2024.

[19] R. Buil, M. A. Piera, and E. Ginters: "Multi-agent system simulation for urban policy design: Open space land use change problem," *Int. J. Model., Simul., Sci. Comput*, vol. 7, no. 2, art. n. 1642002, 2016. Available: https://doi.org/10.1142/S1793962316420022

[20] D. Belytskyi, R. Yermolenko, K. Petrenko, and O. Gogota, "Application of machine learning and computer vision methods to determine the size of NPP equipment elements in difficult measurement conditions," *Machinery and Energetics*, vol. 14, no. 4, pp. 42-53, 2023. Available: https://doi.org/10.31548/machinery/4.2023.42

[21] A. Koldovskyi, "Architectural frameworks for financial transformation in Ukraine," *Dev. Manag*, vol. 23, no. 2, pp. 25-37, 2024.

[22] Y. Degawa, T. Kolzumi, T. Nakamura, R. Shioya, J. Kadomoto, H. Irie, and S. Sakai, "A principal factor of performance in decoupled front-end," *IEICE Trans. Inf. Syst*, vol. 106, no. 12, pp. 1960-1968, 2023.

[23] O. Bezshyyko, A. Dolinskii, K. Bezshyyko, I. Kadenko, R. Yermolenko, and V. Ziemann, "PETAG01: A program for the direct simulation of a pellet target," *Comp. Phys. Commun*, vol. 178, no. 2, pp. 144-155, 2008. Available: https://doi.org/10.1016/j.cpc.2007.07.013

[24] P. Mwiinga, "Privacy-preserving technologies: Balancing security and user privacy in the digital age," *Int. J. Sci. Res. Publ*, 2023. [Online]. Available: https://zenodo.org/records/10406538

[25] K. Khan, *Unlocking Cloud-Native Potential: Building Scalable, Resilient Apps*. Berlin: Kindle Direct Publishing, 2023.

[26] R. Davis, *A Deep Learning and Signal Processing Architecture Using Frequency-Encoded RF Photonics*. Cambridge: Massachusetts Institute of Technology, 2022.

[27] K. Lawal, T. Olaniyi, and R. Gibson, "Leveraging real-world data from IoT devices in a fog-cloud architecture for resource optimisation within a smart building," *Appl. Sci*, vol. 14, no. 1, p. 316, 2023.

[28] E. Blessing and H. Klaus, "Ensuring the security and privacy of data collected through face covering detection," 2023. [Online]. Available: https://www.researchgate.net/publication/377159729_Ensuring_the_security_and_privacy_of_data_collected_through_face_covering_detection

[29] S. V. Symonenko, V. V. Osadchyi, S. O. Sysoieva, K. P. Osadcha, and A. A. Azaryan, "Cloud technologies for enhancing communication of ITprofessionals," *CEUR Workshop Proceed*, vol. 2643, pp. 225-236, 2020.

[30] D. Recupero, D. Dessì, and E. Concas, "A flexible and scalable architecture for human-robot interaction," in *Proc. 15th Eur. Conf. Ambient Intell.*, Cham, Switzerland, 2019, pp. 311-317.

[31] Y. Perera and D. Jayasuriya, "Enhancing the front-end web applications performance using design patterns and microservices based architecture," 2023. [Online]. Available: https://doi.org/10.13140/RG.2.2.36067.53286

[32] O. Nikulina and K. Khatsko, "Method of converting the monolithic architecture of a front-end application to microfrontends," *Bull. Natl. Tech. Univ. "KhPI" Ser. Syst. Anal. Manag. Inf. Technol*, vol. 10, no. 2, pp. 79-84, 2023.

[33] A. Pant, "Importance of data security and privacy compliance," *Int. J. Res. Appl. Sci. Eng. Technol*, vol. 11, no. 11, pp. 1561-1565, 2023.

[34] A. I. Leonow, M. N. Koniagina, S. V. Petrova, E. V. Grunt, S. Y. Kerimkhulle, and V. G. Shubaeva, "Application of information technologies in

marketing: Experience of developing countries,” *Espacios*, vol. 40, no. 38, 2019. Available : http://www.revistaespacios.com/a19v40n38/a19v40n38p24.pdf

[35] S. Kerimkhulle, Z. Dildebayeva, A. Tokhmetov, A. Amirova, J. Tussupov, U. Makhazhanova, A. Adalbek, R. Taberkhan, A. Zakirova, and A. Salykbayeva, “Fuzzy Logic and Its Application in the Assessment of Information Security Risk of Industrial Internet of Things,” *Symmetry*, vol. 15, no. 10, art. n. 1958, 2023. Available: https://doi.org/10.3390/sym15101958

[36] N. Paraskevopoulos, F. Sebastiano, C. Almudever, and S. Feld, "SpinQ: Compilation strategies for scalable spin-qubit architectures," *ACM Trans. Quantum Comput*, vol. 5, no. 1, p. 4, 2023.

[37] S. Naqvi and A. Mohsin, "Using Direct Acyclic Graph (DAG) based Distributed Ledger for a secure and scalable Internet of Things (IoT) architecture," 2023. [Online]. Available: https://uow.edu.pk/ORIC/MDSRIC/Publications/8th%20MDSRIC-148.pdf

[38] T. Krishnappa, "User awareness of security and privacy in social networking sites," *Int. J. Eng. Appl. Sci. Technol*, vol. 8, no. 5, pp. 38-54, 2023.