**NATIONAL RESEARCH UNIVERSITY HIGHER SCHOOL OF ECONOMICS**

**FACULTY OF COMPUTER SCIENCE, ECONOMICS, AND SOCIAL SCIENCES**

**PROJECT PROPOSAL**

**Developing a Highly Scalable and Fault-Tolerant Service for Unstructured Document Management**

**PERM**

**2025**

The purpose of this work is to develop a comprehensive solution to the challenges associated with handling high volumes of unstructured documents while ensuring that workflow stability and scalability are maintained within corporate environments. The primary objective is to design a system that can seamlessly manage vast amounts of unstructured data—such as text documents, images, PDFs, emails, and other formats—while ensuring that the workflow not only meets but exceeds the dynamic needs of modern enterprises. In achieving this, the solution must be robust enough to handle fluctuating data volumes, integrate with existing enterprise systems, and provide high availability and reliability, all while being scalable to accommodate future growth and increasing complexity.

This work also seeks to address the critical aspects of performance, security, and fault tolerance. As businesses increasingly rely on the management of unstructured documents for various processes—ranging from compliance documentation to customer communications—it becomes essential to ensure that these systems can handle large-scale document ingestion, indexing, retrieval, and real-time processing without compromising data integrity. Additionally, the solution must support efficient querying, metadata extraction, and seamless integration with analytics engines, all while providing secure access controls and monitoring capabilities.

Furthermore, the solution must be flexible enough to integrate with various internal and external systems and meet specific workflow requirements as outlined during architectural planning and enterprise requirement gathering phases. These include considerations such as data privacy regulations, authentication methods, and seamless user interface design for both internal users and external consumers accessing the services.

Introduction

**Background.** The exponential growth of digital information has led to an overwhelming surge in unstructured data, encompassing diverse formats like text documents, images, audio files, videos, emails, social media content, and sensor readings. This heterogeneity and sheer volume presents substantial challenges for organizations striving to manage their data efficiently. Traditional relational database systems are often inadequate for this task, necessitating specialized services tailored for the complexities of unstructured data. This work delves into the key considerations and architectural principles crucial for developing a robust, scalable, and fault-tolerant platform for an enterprise-level unstructured document management.Structured data thrives in predefined database formats; however, flexible content requires comprehensive planning since storage, retrieval, and operational models differ significantly. A well-designed system must efficiently balance functionality across different platforms while ensuring fault tolerance and high availability for critical business operations.

Developing a data-intensive, mission-critical solution demands expertise at both the planning and implementation stages, including long-term maintenance strategies and end-of-life considerations. Understanding trade-offs early in the design process is crucial when selecting key infrastructure platforms, balancing costs, risks, and returns. Considerations include hosting and outsourcing decisions, incident recovery setups, and disaster recovery (DR) planning to mitigate service disruptions. Beyond performance metrics like latency and throughput, modern systems must comply with stringent security and authentication requirements, adhering to legal and regulatory standards across industries.

A well-documented architectural plan provides the necessary strategic overview, helping developers anticipate edge cases and proactively address complexities. This reduces technical debt and costly rework caused by rushed implementation decisions under deadline pressures. Proper preparation ensures that enterprise solutions align with business goals, remain cost-effective, and maintain high reliability throughout their lifecycle.

The objective of this project is to develop a solution for handling high volumes of unstructured documents while ensuring workflow stability and scalability within corporate environments. Through in-depth analysis, we aim to define precise project requirements and anticipate potential challenges that must be addressed to achieve a robust final product.This work focuses on creating an enterprise-level document management solution that will handle large-scale unstructured data. The analysis excludes small-scale personal document storage amounts and consumer-oriented cloud storage services. Instead, final product is directed toward business-critical environments where scalability, fault tolerance, and security compliance are primary concerns. Talking about the professional significance this solution holds significant value for organizations aiming to enhance document management efficiency in enterprise environments. By addressing key challenges such as scalability, fault tolerance, and security compliance, the proposed solution contributes to reducing operational risks and improving information retrieval processes. Furthermore, the findings of this study can guide IT departments and decision-makers in selecting the most suitable technological frameworks for large-scale document processing.

Literature review

While creating a service to for structuring unstructured documents can be considered quite an easy task, making the final edition of the system be fault tolerable and scalable enough for an enterprise-level can become a concerning task.

A work by Mihai Baboi et al. (2019) suggest using microservice architecture pointing out their experience based on creating asynchronous applications. Such tasks require high rate of fault tolerability in order to perform correctly, what sounds astounding at the very least. Their method choice is supported by Nils Masuch et al. (2021) in their work about mixing approaches of kubernetes with the microservices in order to achieve a cloud-native computing system, which makes it incredibly easier to scale. This work proves the capabilities of the microservice approach and invokes thoughts of considering it as a most suitable method.

On the other hand article by Priti Kumari and Parmeet Kaur (2022) about cloud computing provides fascinating results of using cloud systems in maintaining complex systems. Their work proves the progress of the cloud systems in providing amenities for maintenance processes in a longshot. Paper written by Krishna Kntikiran Pasupuleti et al. (2018) illustrates the scalability features of cloud systems for storing corporate databases. This work makes cloud computing a great competitor to a microservice approach.

Beyond architecture, processing methodologies significantly impact system performance.AI methods have been rapidly changing and evolving in terms of operating speed. For example the paper by Ciprian-Octavian et al. (2023) provides an in-depth analysis on the easements by using new AI text analyzing pre-processioning methods in order to ease the load on the systems. This approach shows great result in storing and structuring documents by hastening the process and removing rudiment steps. Another work by Rares Vasilescu (2022) suggests using AI contend management techniques for easing the post-process of the contend management algorithms. These authors provide great examples on using the new AI not only optimizing the methods, but adds to the maximum efficiency of the fault tolerance metrics as well.

Big data approaches can also play a crucial role in making a system much faster and more fault-tolerable due to its versatility and complexity. An article by Miruna Stoicescu et al. (2012) provides a great example in using evolvable systems in analytical departmens in frequently changing areas. Their work reaffirms the need in using evolvable bigdata methods in the fields with enormous data flow. To support this idea a work by Andrey Demichev, Alexander Kryukov (2012) on usage of evolving systems in blockchain companies with unimaginable data flow shows incredible fail-safety perfomance. These studies validate the applicability of big data mechanisms in large-scale document management solutions due to their versatility.

After reading all of the articles and analyzing high points of the approaches it is sure that it will be much more efficient to create a microservice architecture with usage of big data methods to ensure great scalability and fault tolerability with less expenses on the computer power.

Methods.

For developing a highly scalable and fault-tolerant service for unstructured document management, this project utilizes a combination of modern technologies and methods that ensure both stable system operation under heavy load and quick recovery in case of failures. Creating such a service requires efficient data storage, easy accessibility, scalability support, data security, and guaranteeing high fault tolerance. These goals are achieved through the use of the R or Elixir programming languages, microservice architecture, K3S clusters, custom-built S3 storage, and monitoring systems with Zabbix and Prometheus.

1. Microservice Architecture and the Use of R or Elixir

One of the foundational solutions to be used in this project is the microservice architecture. This approach divides the system into multiple independent services, each responsible for a specific part of the business logic. Microservices communicate with each other via clearly defined APIs, simplifying the scaling of the system, as well as its modification. This design also improves fault tolerance, as the failure of one service does not directly affect the operation of the others.

For the development of services, the programming languages R and Elixir will be used, ensuring high performance and ease of working with large data volumes. R is particularly well-suited for data analysis and statistical processing, which is useful in the document indexing process and query handling. With R, it is possible to efficiently handle large data sets, integrate various analytical and machine learning models to improve search and document classification quality.

Elixir, built on the Erlang virtual machine, is an excellent choice for building high-performance, fault-tolerant services. It supports scalability and parallel processing of requests, allowing for efficient handling of numerous simultaneous user requests. Additionally, Elixir offers functional programming features that simplify the development of distributed applications and their maintenance.

1. K3S Clusters and Scalability Management

A key component of the infrastructure is the use of K3S – a lightweight version of Kubernetes (K8S). Kubernetes is widely used for orchestrating containerized applications, but for smaller to medium-sized projects, using K8S can be overkill. K3S addresses this issue by providing the same orchestration capabilities but in a more compact and lightweight manner, which is ideal for developing scalable applications with lower resource consumption.

K3S enables the creation of a cluster consisting of multiple servers, which can act as nodes for data storage and processing. Unlike traditional Kubernetes, K3S does not require significant amounts of memory or processing power, making it a perfect solution for cloud environments or physical servers with limited resources.

The primary task K3S addresses in this project is enabling dynamic system scaling. As the project progresses, the service can automatically add new nodes to the cluster, allowing it to handle increased data volumes and higher loads effectively. Importantly, K3S also supports real-time configuration adjustments, enabling system parameters to be modified without stopping services, which significantly increases fault tolerance.

1. Custom S3 Storage for Document Management

A custom S3-compatible object storage solution will be used for storing and managing unstructured documents. This storage system enables efficient management of large data volumes by providing an API for file upload, download, and indexing.

The custom S3 storage offers several advantages:

Scalability: Documents can be stored in a distributed environment, allowing for easy expansion of storage capacity based on the project’s needs.

High Availability: The system ensures data access from multiple nodes, increasing fault tolerance, as data is not lost if one node fails.

Flexibility: The storage supports various data types, making it ideal for managing unstructured documents.

By building a custom S3 storage solution, the project ensures full control over the data management process, ensuring security and providing the necessary features for document handling.

1. Monitoring and Performance Management with Zabbix and Prometheus

To ensure fault tolerance and system monitoring, a combination of Zabbix and Prometheus tools will be used. These systems allow for tracking the status of servers and services while gathering performance metrics to analyze system behavior under load.

Zabbix is a monitoring system that enables real-time tracking of the infrastructure’s components. With Zabbix, alerts can be configured to notify administrators in case of failures or abnormal behavior, such as sudden spikes in CPU usage or unstable service operation. Zabbix also integrates with other security and monitoring systems for enhanced protection and quick incident response.

Prometheus is a monitoring system designed for containerized applications and microservices. It collects metrics from various sources and provides real-time information on service health. Prometheus allows for detailed monitoring of each microservice, offering insights into their status and performance.

Both systems not only help ensure high fault tolerance but also enable performance analysis to optimize the system further.

1. Fault Tolerance and Failure Recovery

A key aspect of developing a highly scalable system is ensuring fault tolerance. To address this, the project will implement automatic load redistribution and service recovery in case of failures. The system must be capable of automatically restarting services and restoring operations with minimal data loss.

Using the R language for query processing and analytics will enable asynchronous system operations and improve fault tolerance through parallel data processing. K3S clusters provide built-in mechanisms for automatically restarting containers in case of failure, minimizing system downtime and ensuring continuous operation.

Thus, to build a highly scalable and fault-tolerant service for unstructured document management, a combination of technologies is utilized, including microservice architecture, K3S clustering, custom S3 storage, and monitoring tools like Zabbix and Prometheus. These technologies ensure high performance, data security, and fault tolerance, as well as flexible scaling depending on the project’s needs.

Results Anticipated

At the moment, software for the project has already been chosen and listed to fulfill all of the necessary requirements. Based on these things listed following results are expected:

1. Application will have a tolerability rate of 99.999%.
2. Application will have a microservice architecture.
3. Application will have JWT tokenisation and encrypted keys in a database.
4. Database of an application will be encrypted.
5. Wake up time will be less then 30 seconds for the whole infrastructure of a project.
6. Application will store and distribute the documents depending on the allowed categories for groups of users.

References

· Baboi, M., Iftene, A., & Gîfu, D. (2019). Dynamic Microservices to Create Scalable and Fault-Tolerant Architecture. Procedia Computer Science, 164, 317-324. DOI: 10.1016/j.procs.2019.09.271.

· Masuch, N., Kirst, D., & Schill, A. (2021). Enabling scalable and fault-tolerant multi-agent systems by utilizing cloud-native computing. Software and Systems Modeling, 20(3), 853-870. DOI: 10.1007/s10458-020-09489-0.

· Pasupuleti, K. K., Klots, B., Nagarajan, V., Kandukuri, A., & Agarwal, N. (2018). High Availability Framework and Query Fault Tolerance for Hybrid Distributed Database Systems. Proceedings of the 2021 ACM Symposium on Cloud Computing, 120-130. DOI: 10.1145/3511808.3557086.

· Kumari, P., & Kaur, P. (2018). A survey of fault tolerance in cloud computing. Journal of King Saud University - Computer and Information Sciences, 30(2), 141-153. DOI: 10.1016/j.jksuci.2018.09.021.

· Sabuhi, M., Musilek, P., & Bezemer, C.-P. (2020). Micro-FL: A Fault-Tolerant Scalable Microservice-Based Platform for Federated Learning. Future Internet, 16(3), 70. DOI: 10.3390/fi16030070.

· Auer, C., Dolfi, M., Carvalho, A., Berrospi Ramis, C., & Staar, P. W. J. (2022). Delivering Document Conversion as a Cloud Service with High Throughput and Responsiveness. arXiv preprint arXiv:2206.00785.

· Stoicescu, M., Fabre, J.-C., & Roy, M. (2012). Experimenting with Component-Based Middleware for Adaptive Fault Tolerant Computing. arXiv preprint arXiv:1204.1232.

· Luo, Y., Li, B., Li, D., & Li, Z. (2012). Continuous Disaster Tolerance in the IaaS Clouds. Proceedings of the 32nd IEEE International Conference on Distributed Computing Systems, 729-738. DOI: 10.1109/ICDCS.2012.45.