CloudCord - Research Plan

Stefan-Nikola Stanev

2024

Contents

1 Problem **2**

2 Opportunity **3**

3 Research questions **4**

3.1 Main question 4

3.2 Sub-Questions 4

4 Research Methods **5**

5 Deliverables **6**

6 Planning **7**

Chapter 1

Problem

In today's digital landscape, real-time communication platforms like Discord and Slack are essential for both enterprise and community collaboration. However, existing solutions often face scalability issues, security vulnerabilities, and limited cloud-native integration. Enterprises require a highly available, secure, and scalable communication system that supports text, voice, and video while ensuring compliance with security best practices (e.g., OWASP, GDPR) and seamless DevOps integration.

CloudCord addresses these challenges by leveraging microservices, cloud-native deployment, and real-time data processing to create a reliable, secure, and scalable communication platform tailored for enterprise use.

Chapter 2

Opportunity

With the growing demand for secure, scalable, and cloud-native communication platforms, there is an opportunity to build a modern alternative that integrates real-time messaging, voice, and video while ensuring enterprise-level security, compliance, and DevOps automation.

CloudCord offers a flexible, self-hosted, and cloud-integrated solution that can scale dynamically, meet strict security standards (e.g., OWASP, GDPR), and provide high availability through microservices and Kubernetes. By focusing on cloud-native best practices, distributed architecture, and CI/CD automation, CloudCord positions itself as an enterprise-ready alternative to existing platforms, catering to both businesses and communities.

Chapter 3

Research questions

3.1 Main question

How can a cloud-native, microservices-based Discord-like platform be designed and implemented to ensure scalability, security, and real-time performance for enterprise use?

3.2 Sub-Questions

1. **Scalability:** What architectural patterns and cloud-native technologies can be used to ensure high availability and scalability in a microservices-based platform?
2. **DevOps:** How can CI/CD, infrastructure as code, and monitoring ensure reliable deployment, scalability and maintainability for CloudCord?
3. **Cloud Native:** How can CloudCord use cloud-native technologies to improve scalability, reliability, and cost efficiency?
4. **Security:** How can CloudCord integrate security best practices, such as authentication, encryption, and OWASP Top 10 mitigation, to protect user data and minimize vulnerabilities?
5. **Distributed Data:** How can CloudCord securely handle and store data while ensuring GDPR compliance and protecting sensitive information?

Chapter 4

Research Methods

Research Sub-Question	**Library Research**	**Field Research**	**Workshop Research**
Scalability	Review scalability principles	Test scalability in real environments	Design scalable architecture
DevOps	Study DevOps tools and practices	Observe DevOps in production	Design CI/CD pipelines and tools
Cloud Native	Review cloud-native principles	Test cloud technologies in real environments	Prototype cloud-native solutions
Security	Study security best practices	Test security in real systems	Design security measures and prototypes
Distributed Data	Review data storage solutions	Test data storage and compliance	Prototype data storage solutions

Table 4.1: Research Methods for Each Research Sub-Question

Chapter 5

Deliverables

1. **Enterprise software solution**

   Software solution integrating cloud concepts and best practices for enterprise software

2. **Documentation**

   (a) Research plan

   (b) Portfolio

   (c) Other documents in the future

Chapter 6

Planning

Sprint 1: Research and Initial Design (Weeks 1-3)

- Research plan
- Initial design documentation, including architecture diagram and selected tools/technologies.
- Non-functional and functional requirements for CloudCord.

Sprint 2: Local Kubernetes Setup, Microservices, and Portfolio (Weeks 4-6)

- Local Kubernetes setup and configuration.
- Initial microservices (user authentication, basic functionality).
- Portfolio documentation with progress so far.

Sprint 3: Security, Additional Microservices, CI/CD, and Portfolio (Weeks 7-9)

- Security implementation (authentication, encryption, OAuth).
- Additional microservices integrated into the system.
- Working CI/CD pipeline (with automated testing).
- Portfolio with added security features and CI/CD documentation.

Sprint 4: Distributed Data, Final Microservices, and Portfolio (Weeks 10-12)

- Distributed data management features (e.g., cloud storage, data replication).

- Final microservices integrated into the platform.
- Cloud-native integrations (e.g., Kubernetes).
- Portfolio with detailed implementation of distributed data and final microservices.

Sprint 5: Deployment, Scalability, and Portfolio (Weeks 13-15)

- Cloud deployment of CloudCord platform.
- Scalability and performance testing results.
- Updated portfolio with deployment and scalability insights.

Sprint 6: Final Solution, Presentation, and Final Portfolio (Weeks 16-18)

- Fully deployed and functional CloudCord platform.
- Final project report and presentation.
- Completed and polished portfolio with all documentation, research, design, and results.