EUROPEAN COMMISSION
Directorate-General for Communications Networks, Content and Technology
CNECT.E – Future Networks
CNECT.E – Future Networks

Analysis, Design, Architecture, Implementation, and Testing of the First Test Prototype & Beta Prototype Roadmap

1. Introduction

This document provides an overview of the analysis, design, architecture, implementation, and testing of the first test prototype of the FeDiversity project's back-end solution, developed using Nix technology. Additionally, it outlines the roadmap for the beta prototype, which is planned for completion by the end of this year.

2. Analysis of Requirements - Nix backend

2.1. Project Scope and Objectives

The FeDiversity project aims to deliver a secure, reproducible, scalable back-end solution using Nix and NixOS. The primary objectives include:

- Ensuring reproducibility and dependency management
- Facilitating robust software deployment
- Improving security through declarative configurations
- Enabling efficient collaboration across development teams

2.2. User and System Requirements

- **Functional Requirements**
  - Authentication and authorization mechanisms
  - Secure API endpoints
  - Data storage and retrieval using Nix-based configurations
  - Integration with front-end and third-party services
- **Non-Functional Requirements**

    - High scalability and availability
    - Optimized performance for data processing
    - Automated deployment and rollback capabilities
    - Compliance with EU data privacy and security standards

3. System Architecture and Design - Nix backend

3.1. Architectural Overview

The back-end solution follows a microservices-based architecture with a declarative approach enabled by Nix. The core components include:

- **Nix-based Package Management**: Ensures reproducible builds and dependency isolation
- **API Gateway**: Centralized entry point for service interactions
- **Authentication Service**: Implements OAuth2 and token-based authentication
- **Data Persistence Layer**: Uses PostgreSQL with Nix-managed configurations
- **Orchestration and Deployment**: Utilizes NixOps and/or Kubernetes for deployment automation

3.2. Design Principles

- **Modularity**: Services are independently deployable and maintainable
- **Immutability**: Configurations and dependencies are strictly version-controlled
- **Security-first Approach**: Implementing best practices for secure software development

4. Implementation of the First Test Prototype - Nix backend

4.1. Development Tools and Technologies

- **Programming Languages**: Shell scripting
- **Infrastructure Management**: Nix, NixOps
- **Version Control**: Git and Forgejo
- **CI/CD Pipelines**: Nix flakes for automated builds

4.2. Key Implementation Milestones

- Setting up the development environment with Nix Flakes and Packages
- Implementing core authentication and API services
- Configuring database persistence with Nix
- Deploying the prototype on test infrastructure

5. Testing and Validation

5.1. Testing Methodology

- **Unit Testing**: Ensuring the correctness of individual components
- **Integration Testing**: Verifying inter-service communication
- **End-to-End Testing**: Simulating real-world usage scenarios

- **Security Audits**: Conducting vulnerability assessments

5.2. Test Results and Observations

- Successful deployment with Nix ensured reproducibility
- Performance testing highlighted areas for optimization
- Security review identified minor configuration adjustments

6. Roadmap for Beta Prototype Development - Nix backend

6.1. Key Enhancements for the Beta Prototype

- Refinement of API interactions and error handling
- Optimization of database performance
- Enhanced security measures, including role-based access control
- Improved deployment automation with NixOps

6.2. Timeline and Milestones

- **Q1 2025**: User feedback integration and architectural refinements
- **Q2 2025**: Feature enhancements and security improvements
- **Q3 2025**: Performance optimization and stress testing
- **Q4 2025**: Finalizing the beta prototype for release

7. Conclusion

The first test prototype successfully validated core functionalities and deployment strategies using Nix technology. Moving forward, the beta prototype will focus on improving security, scalability, and usability to ensure a robust back-end solution for FeDiversity.