Project: Prototype 4 - Quality Attributes, Part 2

1. Objective

The objective of the third delivery of the project is to **redesign the architecture** and generate a **final prototype** of the software system developed as a project.

2. Requirements

2.1 Functional Requirements

- The functional completeness for the software system must be defined by the team.

2.2 Non-Functional Requirements

- The software system must respond to at least four different **reliability** (**high availability**, **resilience** or **fault tolerance**) scenarios:
    - In scenario 1: the software system must implement the **Replication Pattern**.
    - In scenario 2: the software system must implement the **Service Discovery Pattern**.
    - In scenario 3: the software system must implement the **Cluster Pattern**.
    - In scenario 4: the software system must implement a pattern defined by the team.
- The software system must respond to an **interoperability** scenario.
- **Security scenarios** of Prototype 3 must be ensured again during the architectural redesign to be carried out for reliability.
- **Performance** and **scalability scenarios** of Prototype 3 must be revalidated during the architectural redesign to ensure system reliability. The redesign must explicitly address scalability by **adopting an autoscaling method to handle variable workloads efficiently**. Comprehensive performance testing is required again to confirm that the autoscaling approach meets reliability and scalability objectives.

3. Delivery

3.1. Artifact

- Team
    - Team name.
    - Full names and team members.
- Software System
    - Name
    - Logo
    - Description
- Functional and Non-Functional Requirements
- Architectural Structures

    - Component-and Connector (C&C) Structure
        - C&C View
        - Description of architectural styles and patterns used.
        - Description of architectural elements and relations.
    - Layered Structure
        - Layered View
        - Description of architectural patterns used (if applicable).
        - Description of architectural elements and relations.
    - Deployment Structure
        - Deployment View
        - Description of architectural patterns used (if applicable).
        - Description of architectural elements and relations.
    - Decomposition Structure
        - Decomposition View
        - Description of architectural elements and relations.
- Quality Attributes
    - Security
        - Security scenarios.
        - Applied architectural tactics.
        - Applied architectural patterns.
    - Performance and Scalability
        - Performance scenarios.
        - Applied architectural tactics.
        - Applied architectural patterns.
        - Performance testing analysis and results.
    - Reliability
        - Reliability (**high availability**, **resilience** or **fault tolerance**) scenarios.
        - Applied architectural **tactics**.
        - Applied architectural **patterns**.
    - Interoperability
        - Interoperability scenario.
        - Applied architectural **tactics**.
        - Applied architectural **patterns**.
- Prototype
    - Instructions for deploying the software system locally.

3.2. Submission Format

- The deliverable must be submitted via GitHub (swarch2025ii repository).
- Steps:
    - Use the branch **`prototype-4/<team-name>`** (where **`<team-name>`** = team name assigned in the artifact)
    - The path **project/<team-name>** in the repository must include:
        - The artifact in the **README.md** file.
        - The source code (and other configuration files) of the prototype.

3.3. Deadline

On December 08, 2025, before 23:59:59 UTC-5.

3.3. Presentation

On December 09, 2025, during class time.