Project Description

Web-Based Logical Infrastructure Management System for Industrial ICT Maintenance

General Context

In industrial ICT environments, companies rely on complex infrastructures composed of servers, network equipment, industrial machines, and sometimes IoT devices. These infrastructures are physically distributed across data centers, rooms, racks, and shelves. While this physical structure is important, it often creates difficulties in communication, coordination, and maintenance operations, especially when multiple technical teams are involved.

In many real-world situations, maintenance instructions are transmitted verbally or through informal messages, such as telling a technician to go to a specific server, connect cables in a certain shelf, or manipulate a specific port. This type of communication depends heavily on personal experience and local knowledge, which can lead to misunderstandings, mistakes, and delays. The problem becomes even more critical when new technicians join the team or when interventions must be done quickly to ensure service continuity.

Identified Problem

The main problem addressed by this project is the **lack of a shared logical representation of the company's technical infrastructure**. Most existing approaches focus on the physical layout of equipment, while technicians actually need a **clear, abstract, and unambiguous way to understand where a problem is located and what action is required**.

Because of this gap:

- Instructions are often ambiguous
- Different teams interpret the same issue differently
- Troubleshooting becomes slow and error-prone
- Collaboration between field technicians and central systems is inefficient
- Knowledge remains implicit instead of being documented and shared

---

Inspiration and Concept

This project is inspired by modern infrastructure management systems, particularly the way **Kubernetes** organizes complex systems. In Kubernetes, the physical infrastructure (nodes, pods, containers) is abstracted using logical structures such as namespaces, services, and deployments. This abstraction allows engineers to reason about systems logically without worrying about physical placement.

A similar approach is also used in real telecom environments, such as the internal applications used in Tunisie Telecom service centers, where operators can manage subscribers by moving them logically from one piece of equipment to another without directly interacting with physical cables.

Proposed Solution

The proposed solution is a **web application that represents the ICT infrastructure of a company as a logical hierarchy**, rather than only a physical one. The infrastructure is modeled as navigable pages and components, where each level represents a logical entity such as a zone, system, equipment group, or component.

Instead of telling a technician "go to server A and shelf B," the system allows tasks to be described logically, for example by indicating a specific infrastructure zone, system, and component. This gives the technician a clear abstract understanding of the issue before performing the physical intervention, reducing confusion and errors.

Key Features of the Application

The application provides a centralized platform where technical staff can:

- Navigate through the logical structure of the company's infrastructure
- Visualize equipment and systems as hierarchical components
- Declare and track technical issues linked to logical entities
- Assign maintenance tasks to technicians with precise logical descriptions
- Collaborate through comments and shared task history
- Follow the status of interventions and ensure service continuity

This logical representation acts as a common language between all teams, ensuring that everyone refers to the same structure when discussing problems or solutions.

Technical Architecture

The application is designed as a modern web system:

- **Frontend (Angular):**
  Used to build a clear and interactive interface where infrastructure elements are represented as pages and components. Technicians can easily navigate through zones, systems, and equipment.
- **Backend (Spring Boot):**
  Handles business logic, task management, issue tracking, and user coordination through REST APIs.

- **Database (MongoDB):**
  Stores the infrastructure structure, equipment data, issues, and tasks using flexible schemas that can evolve as the infrastructure grows or changes.

Expected Benefits

By introducing a logical abstraction layer for infrastructure management, the system aims to:

- Reduce misunderstandings during maintenance operations
- Improve troubleshooting efficiency
- Facilitate collaboration between field teams and central systems
- Simplify onboarding of new technical staff
- Improve traceability and documentation of interventions

Conclusion

This project proposes a practical and modern approach to industrial ICT maintenance by separating the **logical understanding of infrastructure** from its physical complexity. By providing a shared, structured, and abstract representation of systems and equipment, the application improves communication, coordination, and reliability in maintenance operations, while remaining inspired by proven real-world systems used in cloud computing and telecommunications.