Daily Report

Jurgen Halili 30/09/2025

Today I completed the initial migration from the legacy hardcoded MCP system to a database-driven approach. Previously, all MCP tools were defined inside a massive 3600-line service file with static switch statements, which made the system unmaintainable. I refactored this by creating a database schema for storing tool configurations, allowing them to be dynamically loaded and managed without code changes.

On the backend, I built full CRUD APIs for MCP tools, enabling creation, editing, updating, and deletion directly via the database. These endpoints include validation and permission checks to ensure tools are properly scoped by user roles. The system now supports dynamic tool loading, which lays the groundwork for transitioning to the new Unified MCP system.

I also implemented a complete frontend UI for managing tools. This includes a new management panel where superadmins can view all tools, create new ones, edit execution details, and disable or delete existing entries. The UI was built with proper form validation, clear status indicators, and integration with the backend APIs. Together, this makes tool management accessible without modifying backend code.

By the end of the day, the system successfully transitioned from a static, hardcoded implementation to a flexible, database-backed system with a fully functional UI and API layer. This migration provides the foundation for the dynamic execution and security framework that I will implement in the next stages.