**Jurnal etapa 2 – Ricu Alexandru Razvan**

## Cerinte rezolvate

In aceasta etapa am realizat containerizarea completa a proiectului folosind Docker si docker-compose. Au fost create trei imagini: una pentru Ollama, una pentru serverul MCP si una pentru ADK-Web impreuna cu agentul. Serverul MCP a fost extins cu un mecanism de autentificare pe baza de token-uri, astfel incat doar agentii autorizati sa poata accesa tool-urile expuse. Toate aceste servicii au fost orchestrate printr-un fisier docker-compose, ceea ce permite pornirea sistemului cu o singura comanda si asigura comunicarea corespunzatoare intre ele.

## Modul de rezolvare

Pentru a containeriza modelul LLM, am creat o imagine dedicata care instaleaza Ollama si incarca modelul gpt-oss:20b necesar agentului. Aceasta imagine expune portul utilizat de serviciul Ollama si permite incarcarea automata a modelului la pornirea containerului.

Serverul MCP a fost mutat intr-o imagine separata care porneste aplicatia pe portul 8001 si foloseste protocolul HTTP Streaming. Pe langa functionalitatile implementate in etapa precedenta, aici a fost adaugat si un mecanism de autentificare. Token-urile sunt verificate la fiecare request, conform documentatiei oficiale a MCP pentru autorizare. Pentru gestionarea autentificarii am utilizat Keycloak, care emite token-uri JWT ce sunt validate in containerul MCP inainte ca un agent sa poata folosi tool-urile serverului.

Pentru interfata si clientul local, a fost creata o imagine ce include ADK-Web si agentul implementat anterior. Agentul este configurat sa foloseasca hostname-ul containerului MCP in locul unei adrese locale, astfel incat comunicarea sa functioneze corect in interiorul retelei Docker.

Toate imaginile au fost integrate intr-un fisier docker-compose ce defineste reteaua interna, dependentele dintre servicii si volumele necesare pentru persistenta modelelor Ollama si a fisierelor gestionate de MCP. Odata

pornit, intregul sistem se conecteaza automat: ADK-Web comunica cu agentul, agentul cu serverul MCP, iar MCP cu modelul Ollama.

**Probleme intampinate si modul de rezolvare**

Cea mai dificila parte a fost configurarea corecta a Dockerfile-urilor. Initial, containerele nu porneau corespunzator, fie din cauza unor rute gresite, fie din cauza unor porturi incorect expuse. Am rezolvat aceste probleme prin testarea individuala a fiecarei imagini si standardizarea structurii directoarelor.

Integrarea autentificarii a fost o alta provocare. Comunicarea cu Keycloak nu functiona corect la inceput, deoarece serverul MCP nu reusea sa valideze token-urile. Am ajustat configuratiile Keycloak (realm, client, permisiuni) si am adaugat o logica suplimentara in middleware-ul MCP pentru a trata cazurile in care Keycloak nu era inca disponibil la pornirea containerelor.

**Concluzii**

Aceasta etapa a proiectului a dus la transformarea arhitecturii initiale intr-un sistem modular si complet containerizat. Am invatat sa construiesc imagini Docker eficiente, sa configurez retele interne in docker-compose si sa integrez un mecanism de autentificare bazat pe token-uri pentru un server MCP. Totodata, am deprins abilitati practice legate de depanare, sincronizarea serviciilor si comunicatia dintre containere.