

ProceduralNexus

1. Descrierea Aplicației și a Funcționalităților Principale

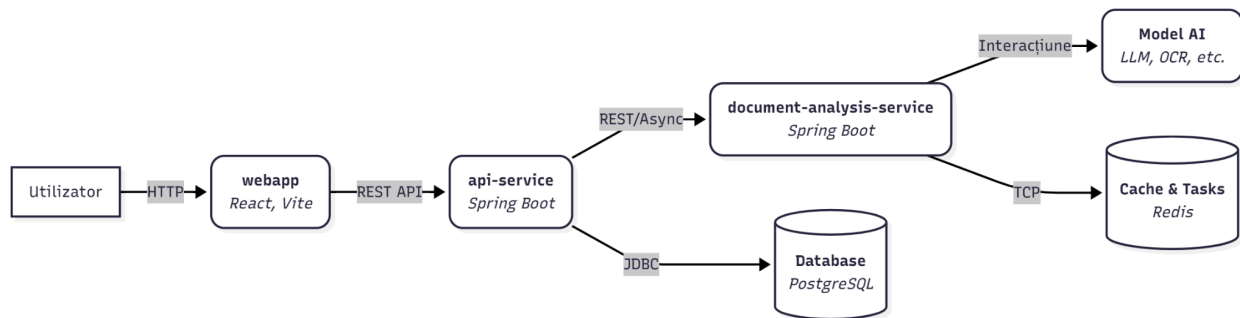
Descriere Generală

ProceduralNexus este o platformă de tip *Intelligent ERP & Document Management System*, concepută pentru a digitaliza și simplifica interacțiunile birocratice dintre utilizatori individuali și organizații. Aplicația servește drept un punct de convergență ("Nexus") unde utilizatorii își pot gestiona cariera și documentele, iar organizațiile (prin HR și Admini) pot administra proceduri interne, recrutare și fluxuri de semnare. Elementul diferențiator este integrarea unui **Agent AI avansat** care nu doar analizează datele, ci poate interacționa cu sistemul pentru a asista utilizatorii în timp real.

Funcționalități Principale

- **Ecosistem Organizațional (Organization Market):** O piață digitală unde utilizatorii fără afiliere pot descoperi organizații și pot aplica, iar companiile își pot gestiona vizibilitatea și procesele de recrutare.
- **Dashboard-uri Specializate:** Interfețe dedicate pentru diferite roluri — **User** (gestionare profil și documente personale), **HR** (management angajați și proceduri), și **Admin** (configurare sistem).
- **Managementul Inteligent al Documentelor:** Stocare centralizată cu funcționalități de vectorizare, permițând căutări semantice și extragere de informații ("Insights") din volume mari de texte.
- **Agent AI Activ (Q&A & Actions):** Un asistent bazat pe AI care răspunde la întrebări din regulamentele interne (prin RAG - Retrieval Augmented Generation) și, crucial, **poate executa acțiuni prin API** (ex: "Semnează cererea de concediu" sau "Caută angajații din departamentul IT").
- **Automatizare și E-Signature:** Fluxuri de lucru automate pentru aprobări și integrare nativă a semnăturii electronice.

2. Diagrama de Arhitectură de Nivel Înalt



3. Descrierea Arhitecturii

Arhitectura ProceduralNexus este una modernă, distribuită, bazată pe microservicii containerizate, punând accent pe scalabilitate și pe capacitatea AI-ului de a opera ca un actor activ în sistem.

Componentele principale sunt:

1. WebApp (Frontend):

- Dezvoltată cu **React și Vite**, oferă o experiență de utilizare rapidă și reactivă.
- Include rute protejate și dashboard-uri dinamice (User, HR, Admin).
- Comunică exclusiv cu **API Service** prin request-uri HTTP securizate.

2. API Service (Backend Core):

- Serviciul central (monolit modular sau microserviciu principal) construit cu **Spring Boot**.
- Gestionează logica de business critică: Autentificare (JWT), CRUD Organizații, Management Utilizatori și Logica de E-Signature.
- Expune o serie de **Endpoint-uri REST** standardizate.
- Servește ca "sursă de adevăr" și mediator între baza de date, frontend și serviciul de AI.

3. AI Agent Service (Backend AI & Orchestrator):

- Un microserviciu specializat (Python/FastAPI sau Spring Boot) care gestionează logica de Inteligență Artificială.
- **Capabilitate Duală:**
 - **Procesare:** Primește documente pentru vectorizare și analiză.
 - **Agent Activ (API Consumer):** Are capacitatea de a apela înapoi **API Service** (Backend Core). Acest lucru îi permite AI-ului să verifice date în timp real (ex: "Există userul X?") sau să execute acțiuni la comanda utilizatorului (ex: să apeleze endpoint-ul de creare a unei notificări sau de actualizare a statusului unui task).
- Folosește un **LLM** (Large Language Model) integrat cu o bază de date vectorială pentru context.

4. **PostgreSQL (Baza de date Principală):**

- Stochează datele relaționale persistente: utilizatori, structura organizațiilor, metadatele documentelor și log-urile de audit.

5. **Redis (Cache & Vector Store & Queues):**

- Folosit pentru:
 - **Caching:** Optimizarea sesiunilor și a datelor accesate frecvent.
 - **Message Broker:** Coordonarea sarcinilor asincrone între API Service și AI Service.
 - **Vector Store (Optional):** Stocarea vectorilor generați din documente pentru căutări rapide de tip RAG.

6. **Infrastructure (Docker):**

- Întregul mediu este definit prin **docker-compose**, asigurând consistența între mediul local de dezvoltare (devcontainer) și producție.

4. Lista Membrilor Echipei și Împărțirea Sarcinilor

- **Frontend:** Cătălina Constantin și Andrea Iuhasz
- **Backend Core & API:** Elvis Hărdălău și Dragoș Marogel
- **Micro-serviciu AI, Integrări & DevOps:** Vlad Vălean