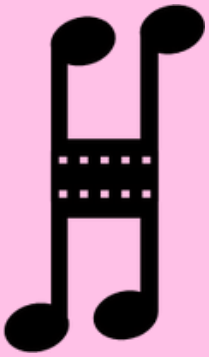
MARTIE 2025

DOCUMENT DE PROIECTARE



VIBECHECK

REALIZAT DE: ECHIPA VIBECHECK

MĂDĂLINA ANDRONACHE, DARIA BANU, IOANA IORDACHE, ALEXIA MIHAI, CARLA RUSU

CUPRINS

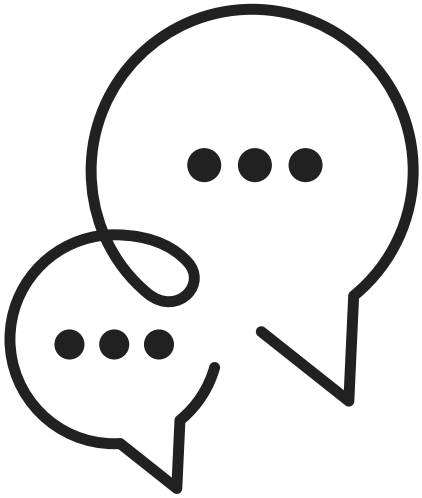
01

SCOPUL SISTEMULUI

VibeCheck este o platformă de messenger care permite utilizatorilor să interacționeze, să partajeze recomandări muzicale și cinematografice, și să creeze comunități bazate pe interese comune. Scopul acestui document este de a defini arhitectura sistemului și de a structura componentele software necesare implementării aplicației.

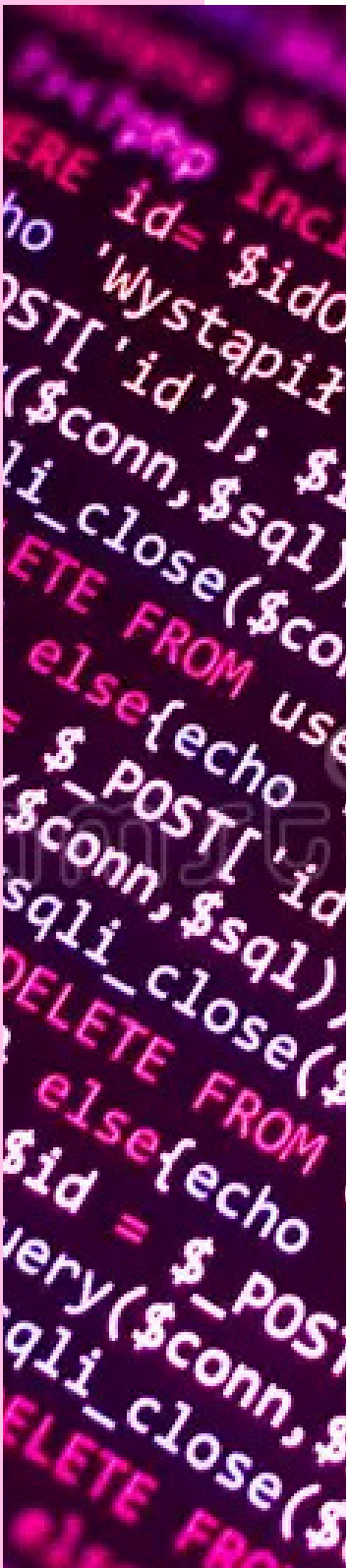
DOCUMENTE DE REFERINȚĂ

1) Specificarea cerințelor software (SRS) - VibeCheck
2) Documente tehnice de integrare API (Spotify API, TMDB API)



02

OBIECTIVE DE PROIECTARE



- O bună organizare a codului
- Separarea clară a subsistemelor și a funcționalităților acestora
- Asigurarea unei latențe reduse și a unei arhitecturi scalabile
- Implementarea unui sistem intuitiv de gestionare a utilizatorilor și a conținutului
- Asigurarea securității și confidențialității datelor utilizatorilor
- Optimizarea performanței pentru a gestiona un număr mare de utilizatori simultan

03

PREZENTARE GENERALĂ

Arhitectura aplicației este bazată pe o separare clasică backend-frontend, având la bază tehnologii moderne pentru asigurarea unei performanțe ridicate și a unei experiențe fluide pentru utilizatori. Această arhitectură modulară facilitează scalabilitatea, securitatea și extensibilitatea aplicației.

- **Frontend** - React.js, utilizat pentru construirea unei interfețe interactive și dinamice. Componentizarea interfeței permite o mentenanță ușoară și o dezvoltare rapidă de noi funcționalități. React oferă un model declarativ pentru construirea interfețelor și permite reutilizarea componentelor UI.

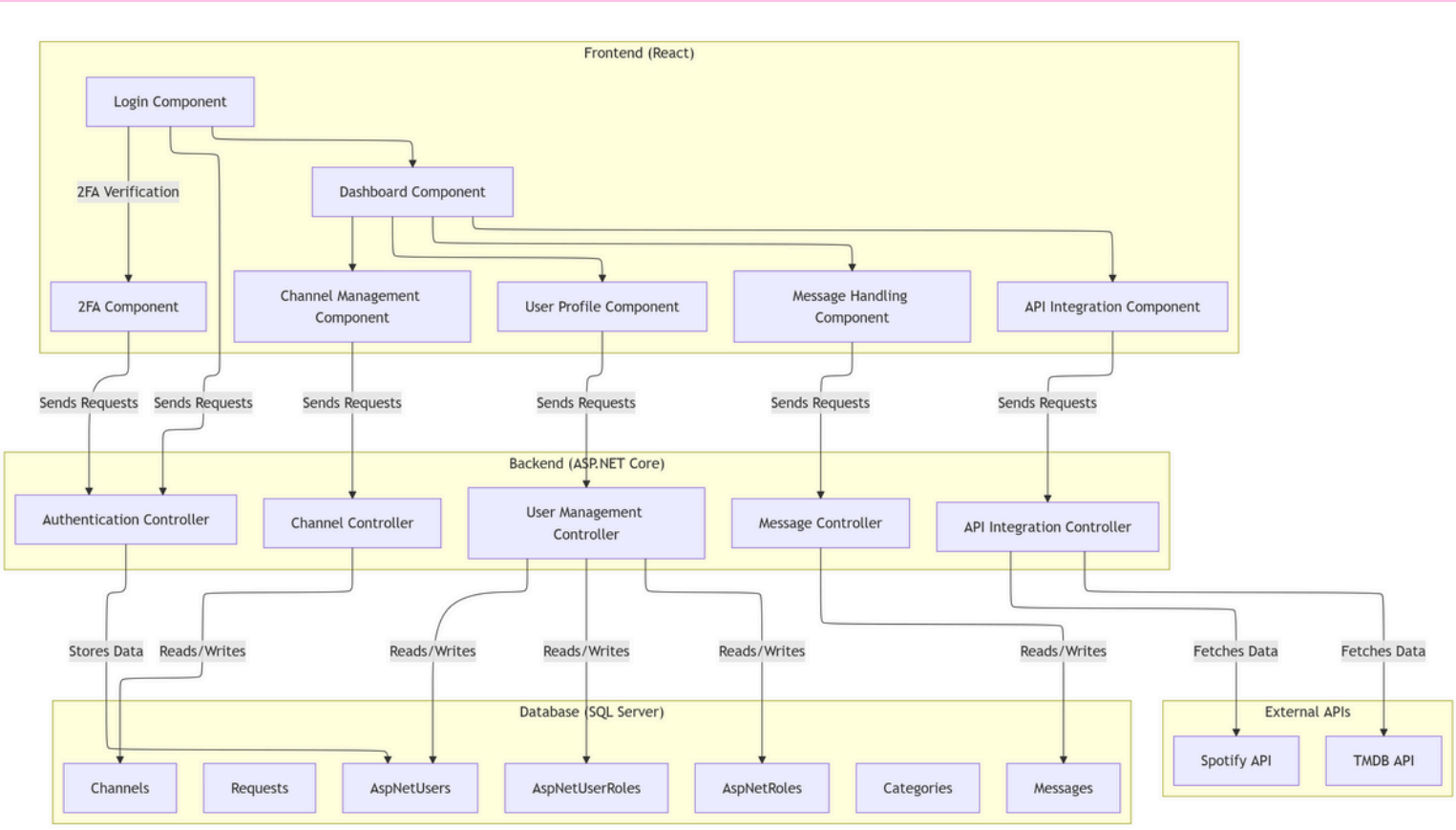
- **Backend** - ASP.NET Core, un framework modern și performant pentru dezvoltarea aplicațiilor backend scalabile. Acesta gestionează cererile venite de la frontend, se ocupă de procesarea datelor și comunică cu baza de date și serviciile externe.

- **Baza de date** - SQL Server, utilizată pentru stocarea datelor persistente ale utilizatorilor, mesajelor și recomandărilor multimedia. Se implementează mecanisme de caching pentru optimizarea interogărilor și reducerea timpului de răspuns.

- **Integrare cu API-uri** - Spotify API și TMDB API sunt utilizate pentru accesarea și gestionarea conținutului multimedia, cum ar fi melodii, albume, filme și seriale. Aceste API-uri permit utilizatorilor să descopere conținut nou și să partajeze preferințele lor în mod dinamic.

03

DIAGRAMA DE COMPONENTE



03

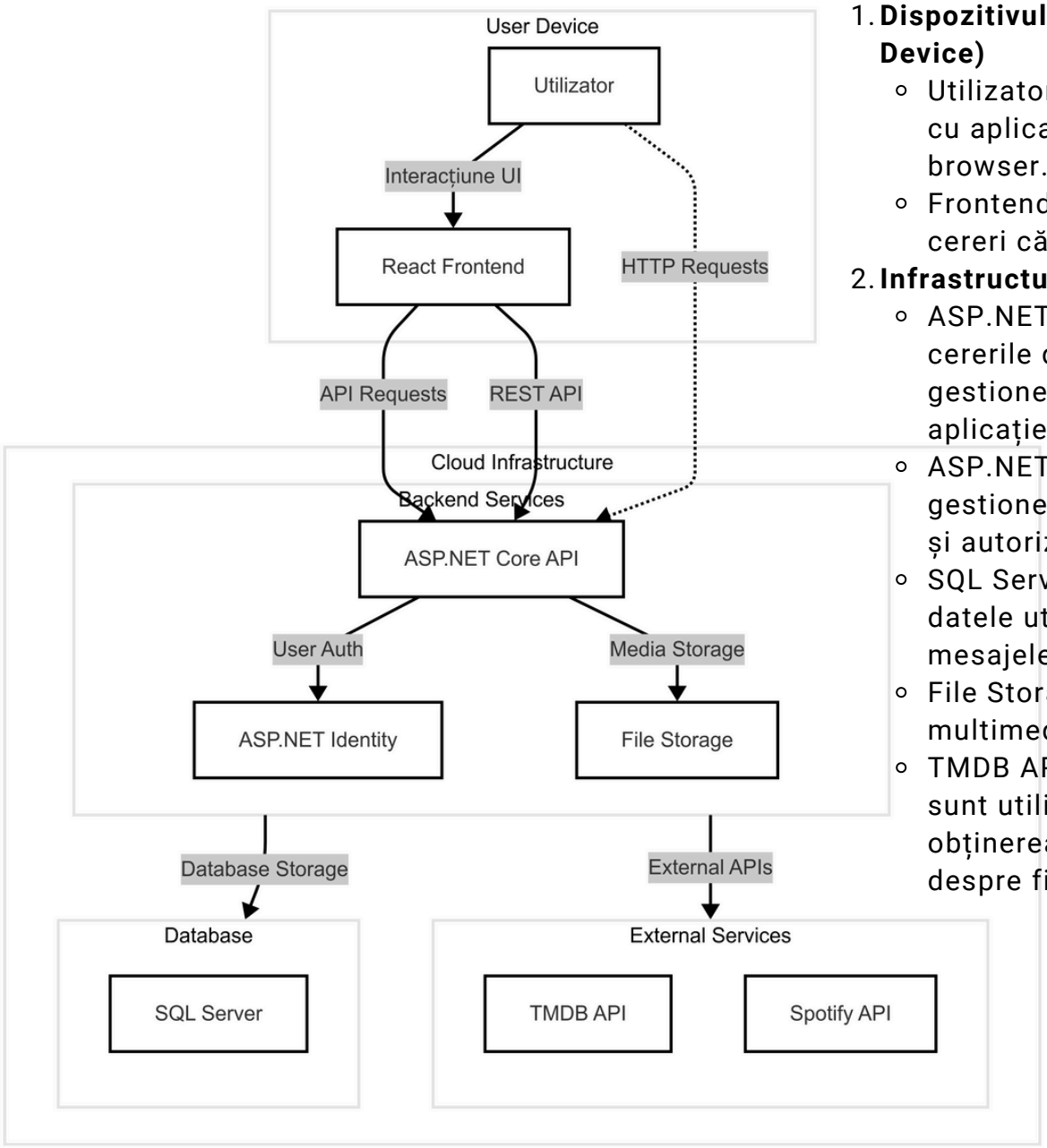
DECOMPOZIȚIA ÎN SUBSISTEME ȘI RESPONSABILITĂȚI

- **Frontend:**
  - UI Component System – Gestionează interfața utilizatorului folosind React
  - State Management – Utilizarea Context API / Redux pentru gestionarea stării
  - API Service Layer – Comunicarea cu backend-ul ASP.NET prin HTTP requests
  - Authentication & Authorization – Gestionarea autentificării utilizatorilor cu JWT și protecția rutelor

- **Backend:**
  - API Gateway – Gestionarea cererilor HTTP primite de la frontend
  - Identity & Authentication – Utilizarea ASP.NET Identity pentru autentificare și autorizare
  - Data Management – Manipularea datelor utilizând Entity Framework Core și SQL Server.
  - Media Handling – Gestionarea fișierelor media și stocarea acestora pe server
  - Integration Services – Conectare la TMDB API și Spotify API pentru obținerea datelor despre filme și muzică

- **Baza de date:**
  - SQL Server Database – Păstrează datele despre utilizatori, mesaje, canale și cereri.
  - File Storage – Depozitarea fișierelor multimedia trimise prin aplicație.

03 DISTRIBUȚIA SUBSISTEMELOR PE PLATFORMA HARDWARE/SOFTWARE (DIAGRAMA DE DISTRIBUȚIE) - V2



1. **Dispozitivul utilizatorului (User Device)**
   - Utilizatorul interacționează cu aplicația React în browser.
   - Frontend-ul (React) trimite cereri către backend.
2. **Infrastructura Cloud - Backend**
   - ASP.NET Core API primește cererile de la frontend și gestionează logica aplicației.
   - ASP.NET Identity gestionează autentificarea și autorizarea.
   - SQL Server stochează datele utilizatorilor, mesajele și canalele.
   - File Storage reține fișiere multimedia.
   - TMDB API și Spotify API sunt utilizate pentru obținerea informațiilor despre filme și muzică.

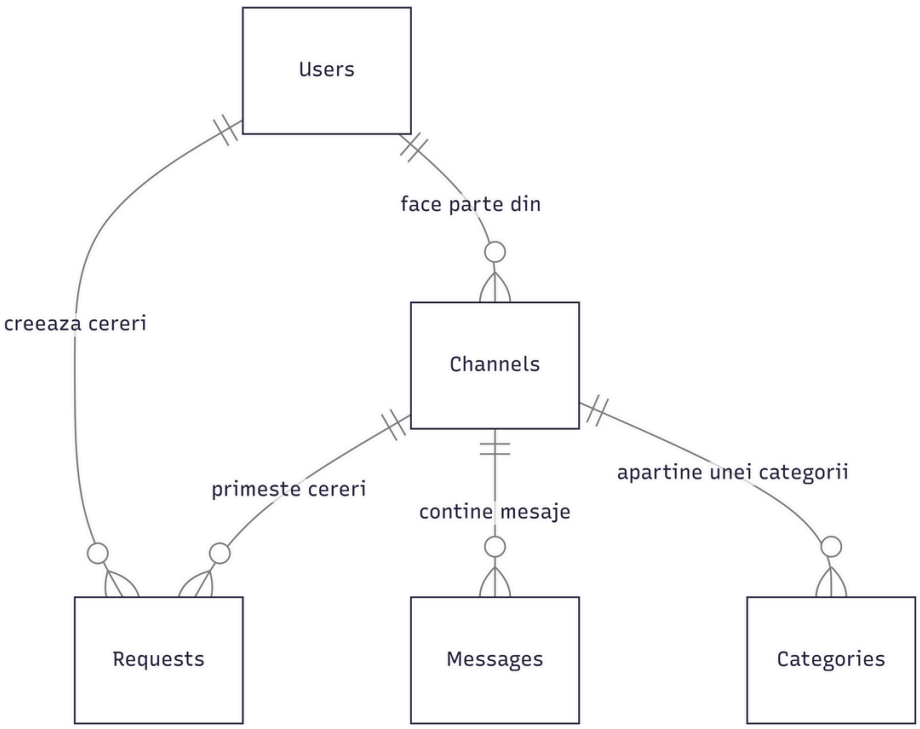
03

MANAGEMENTUL DATELOR PERSISTENTE

O să utilizăm o bază de date, probabil SQL Server, pentru a stoca informațiile esențiale despre utilizatori, mesaje, canale și cereri de alăturare.

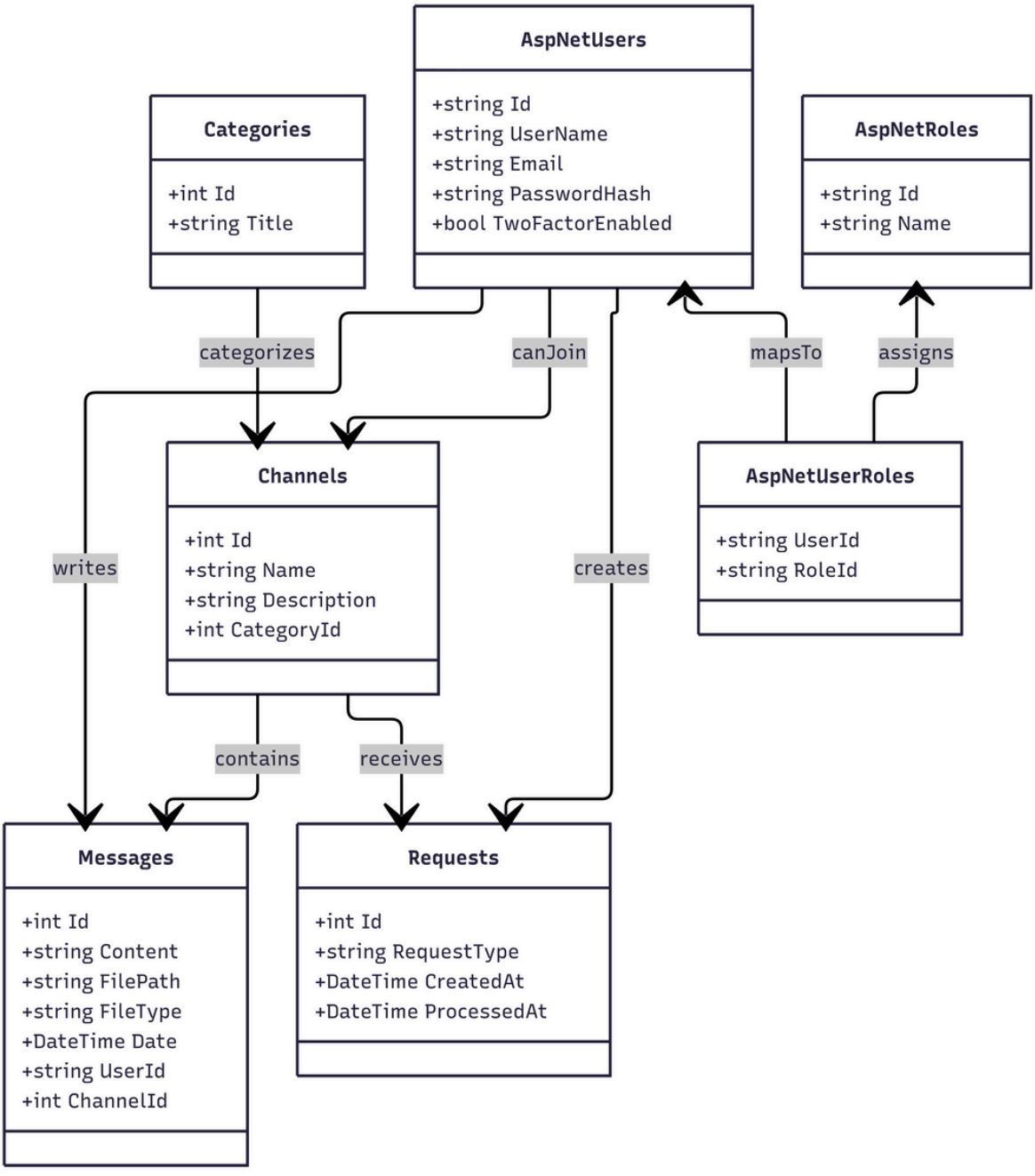
Ne gândim să avem următoarele tabele principale:

- Utilizatori – informații despre conturi și autentificare
- Roluri și permisiuni – mecanisme de control al accesului
- Canale – grupuri de discuție structurate
- Categorii – organizarea canalelor în teme
- Mesaje – conținutul trimis în discuții
- Cererile de alăturare – gestionarea accesului utilizatorilor



03

MANAGEMENTUL DATELOR PERSISTENTE



03

CONTROLUL ACCESULUI UTILIZATORILOR ÎN SISTEM

Controlul accesului în sistem va fi gestionat printr-un mecanism bazat pe roluri și permisiuni, utilizând **ASP.NET Identity**. Se vor defini trei roluri principale, fiecare având niveluri diferite de acces și permisiuni.

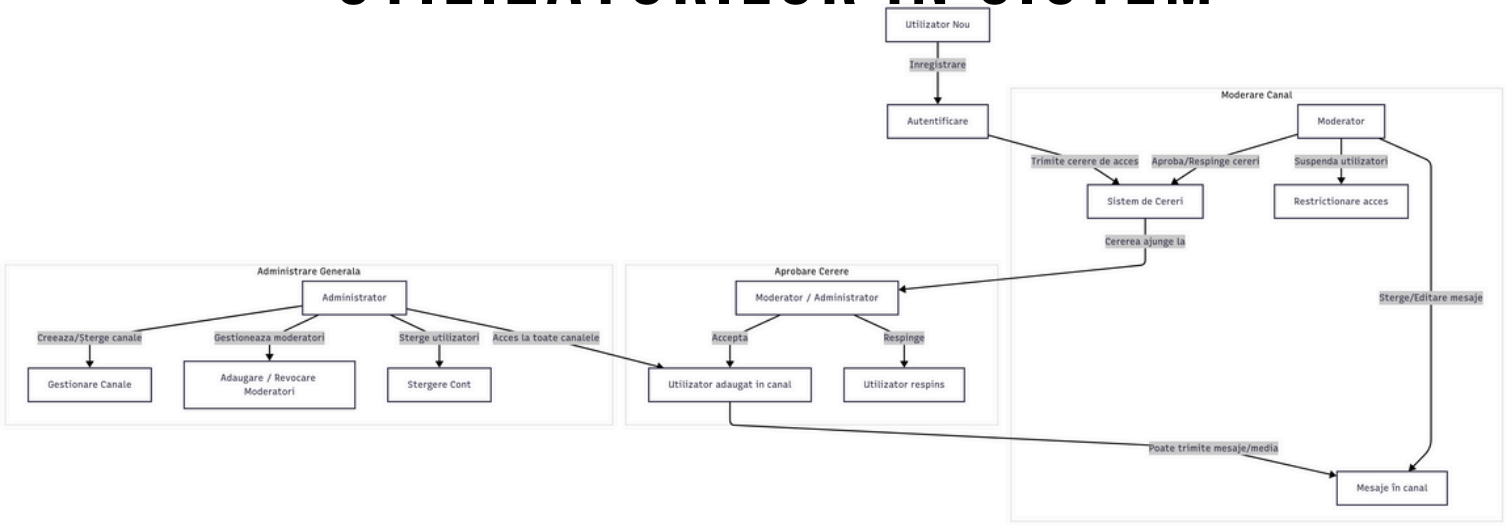
Mecanismul de Control al Accesului

Mecanismul de Control al Accesului

- **Autentificare**: Se face prin email și parolă, cu posibilitate de **2FA** pentru securitate suplimentară.
- **Autorizare**: Se verifică rolul utilizatorului pentru a decide ce acțiuni poate efectua.
- **Permisiuni Granulare**: Moderatorii au puteri limitate la canalele lor, iar administratorii au control total.
- **Gestionare Dinamică**: Un utilizator poate deveni moderator doar pe anumite canale, fără a avea permisiuni globale.

03

CONTROLUL ACCESULUI UTILIZATORILOR ÎN SISTEM



User Normal

- Se poate înregistra și autentifica în platformă.
- Poate cere acces la unul sau mai multe canale.
- Poate trimite mesaje și partaja media în canalele în care a fost acceptat.
- Nu poate modera conținutul sau gestiona utilizatori.

Moderator

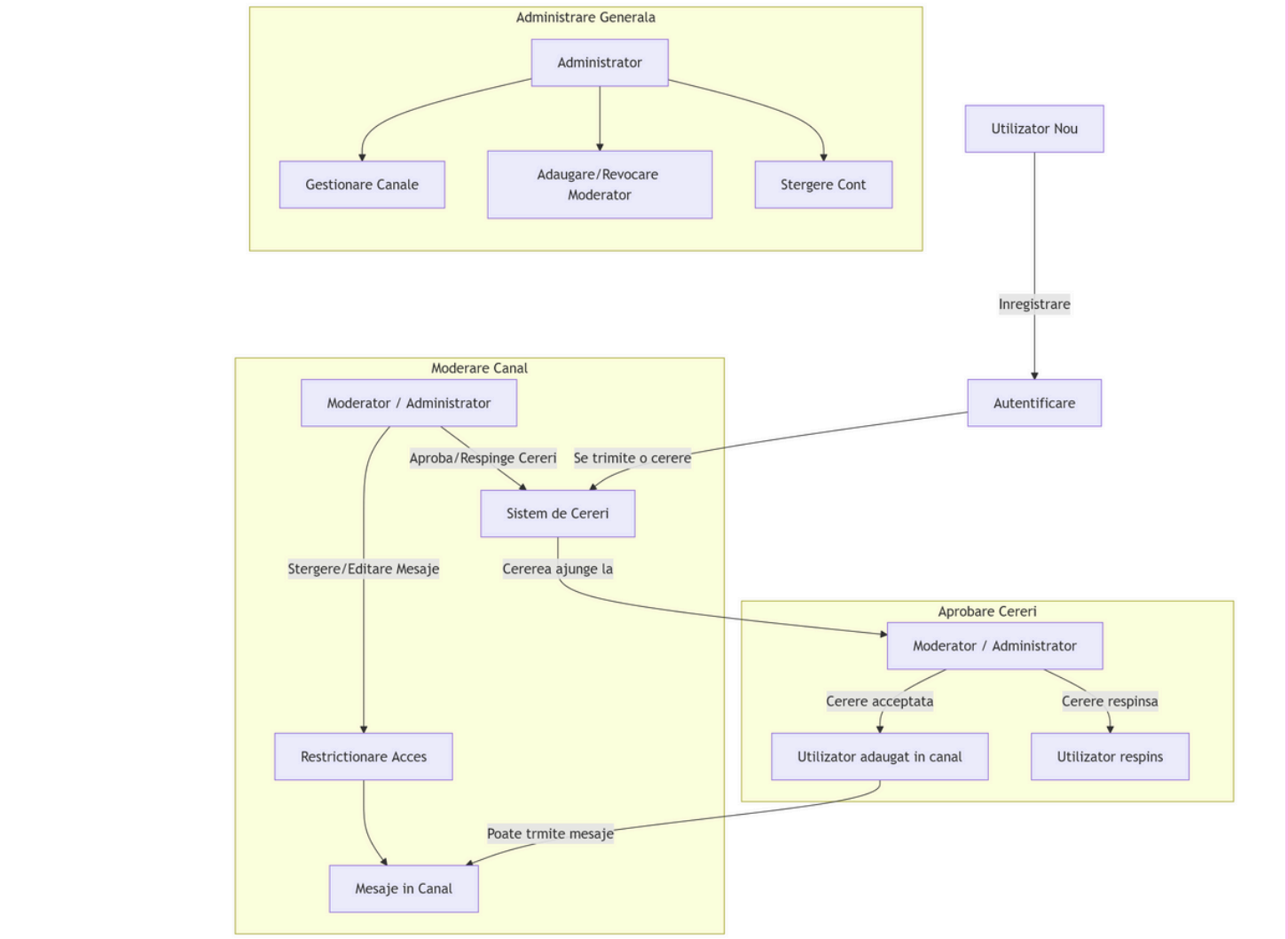
- Are permisii de moderare doar pe canalele pentru care este desemnat.
- Poate aproba sau respinge cereri de alăturare la canalul pe care îl moderează.
- Poate șterge sau edita mesaje din canalul respectiv.
- Poate suspenda utilizatori din canalul său, dar nu îi poate șterge definitiv.

Administrator

- Are control global asupra tuturor canalelor și utilizatorilor.
- Poate crea, edita sau șterge canale.
- Poate numi și revoca moderatori pe orice canal.
- Poate șterge utilizatori din platformă.
- Poate vedea și gestiona toate mesajele și setările aplicației.

03

CONTROLUL ACCESULUI UTILIZATORILOR ÎN SISTEM



03

FLUXUL GLOBAL AL CONTROLULUI (DIAGRAMĂ DE ACTIVITATE)