ROMÂNIA
MINISTERUL APĂRĂRII NAŢIONALE
ACADEMIA TEHNICĂ MILITARĂ „FERDINAND I”

FACULTATEA DE SISTEME INFORMATICE ŞI SECURITATE CIBERNETICĂ
Specializarea: Calculatoare şi sisteme informatice pentru apărare şi securitate naţională

Tehnici de anonimizare a datelor cu caracter personal în aplicațiile medicale

CONDUCĂTOR ȘTIINŢIFIC:
Col. prof. dr. ing.ION BICA

ABSOLVENT:
Stud. plt. adj.Ioana Dragoș

Conţine _____ file
Inventariat sub nr. _____
Poziţia din indicator: ____
Termen de păstrare: _____

BUCUREŞTI
2022

ABSTRACT

Anonymization brings together the set of legal policies, artificial intelligence algorithms and cryptographic or probabilistic procedures, the purpose of which is to remove the ability to identify personal data. The techniques frequently approached for the implementation of an architecture that respects the norms of personal data protection, represent either the masking of data, through substitution operations, permutations, encryptions or statistical calculations, or the generation of an independent data set, which preserves the fundamental characteristics. Which cannot be reached in the original set. The malicious extraction of data from an application that implements a previously detailed architecture can be prevented by complying with the legal rules in force which describe the management of private data and which are based on the considered mathematical formalism of the suggested operations.

This paper aims to exemplify a real data register that can be operated in optimal, secure conditions, which respects the agreements on the processing of personal data and the development of a solution that allows the storage of medical data based on ensuring integrity, authenticity, their confidentiality and their anonymity in order to calculate statistics, probabilities and medical risk factors. The proposed solution will have to ensure the training of a Deep Learning model through a federated learning network of end devices that allows the generation of anonymized differentially-private data set, the creation of a smart contract that manages the data collected on a web platform and persists it within the Ethereum test blockchain network.

REZUMAT

Anonimizarea reunește setul de politici legale, algoritmi de inteligență artificială și proceduri criptografice sau probabilistice, al căror scop este eliminarea capacității de identificare a datelor cu caracter personal. Tehnicile abordate frecvent pentru implementarea unei arhitecturi care respectă normele de protecție a datelor cu caracter personal, reprezintă fie mascarea datelor, prin operații de substituire, permutări, criptări sau calcule statistice, fie generarea unui set de date independent, care păstrează baza fundamentală,caracteristici, care nu pot fi atinse în setul original. Extragerea rău intenționată a datelor dintr-o aplicație care implementează o arhitectură detaliată anterior poate fi prevenită prin respectarea normelor legale în vigoare care descriu gestionarea datelor private și care se bazează pe formalismul matematic considerat al operațiunilor propuse.

Lucrarea de față își propune să exemplifice un registru real de date care poate fi operat în condiții optime, securizate, care respectă acordurile privind prelucrarea datelor cu caracter personal și dezvoltarea unei soluții care să permită stocarea datelor medicale bazată pe asigurarea integrității, autenticității, a acestora. Confidențialitatea și anonimatul acestora au în vedere calcularea statisticilor, probabilităților și factorilor de risc medical. Soluția propusă va trebui să asigure antrenamentul unui model de Deep Learning printr-o rețea de învățare federată de dispozitive finale care să permită generarea unui set de date anonimizate diferențiat-privat, crearea unui contract inteligent care gestionează datele colectate pe o platformă web și persistă în rețeaua blockchain de testare Ethereum.

CUPRINS

LISTĂ DE ABREVIERI

Nr. crt.	Abreviere/ Acronim	Forma completă
1.	GDPR	General Data Protection Regulation
2.	HIPAA	Health Insurance Portability and Accountability Act
3.	PII	Personally identifiable information
4.	DES	Data Encryption Standard
5.	3DES	Triple Data Encryption Standard
6.	RC4	Rivest Cipher 4
7.	RSA	Rivest–Shamir–Adleman
8.	SHA-3	Secure Hash Algorithm 3
9.	SMPC	Secure Multi Party Computation
10.	SecAgg	Secure Aggregation
11.	OpenEMR	Open-Source Electronic Medical Records
12.	FHIR	Fast Healthcare Interoperability Resources
13.	API	Application programming interface
14.	iOS	iPhone Operating System
15.	REST	Representational state transfer
16.	I/O	Input/Output
17.	HTML	HyperText Markup Language
18.	JSX	JavaScript Syntax Extension
19.	DOM	Document Object Model
20.	VDOM	Virtual Document Object Model
21.	SSR	Server-side rendering
22.	SSG	Static Site Generation
23.	JSON	JavaScript Object Notation
24.	RPC	Remote procedure call
25.	HTTP	Hyper Text Transfer Protocol
26.	HTTPS	Secure Hyper Text Transfer Protocol
27.	SHA	Secure Hash Algorithm
28.	PBKDF2	Password-Based Key Derivation Function 2
29.	HMAC	Hash-based message authentication code
30.	PySyft	Python library for secure and private Deep Learning
31.	PyTorch	Python package that provides two high-level features
32.	Python	Programming language
33.	PyGrid	Python utilities to help interface with Sun Grid Engine
34.	OMS	Organizația Mondială a Sănătății
35.	OOP	Object-oriented programming
36.	IDE	Integrated Development Environment

LISTĂ DE FIGURI

LISTĂ DE TABELE

LISTĂ DE ECUAȚII

Nr. Crt.	Indice	Denumire
1.	(2.1)	Acuratețe
2.	(2.2)	Rechemare
3.	(2.3)	Precizie
4.	(2.4)	Scor F1
5.	(2.5)	Minimizarea costului asociat modelului global
6.	(2.6)	Procesul de mascare aditivă

1. INTRODUCERE

Lucrarea prezentă are în vedere dezvoltarea unui registru electronic digital implementat sub forma unei platforme Web care să permită agregarea de date medicale de la pacienți și furnizarea de diagnostice specifice datelor sub formă anonimizată, pentru a nu divulga caracterul personal al datelor și a nu permite identificarea persoanei de la care provin. Aceasta va fi realizată cu ajutorul tehnologiei blockchain pusă la dispoziție de Ethereum, tehnologiei de server-side rendering pusă la dispoziție de framework-ul Next și a librăriei Web3.

Aplicația va furniza utilizatorului, următoarele capabilități:

1. afișarea imaginilor anonimizate și a clasei asociate din cadrul fiecărui diagnostic;
2. vizualizarea datelor privind pacienții care au conferit drepturi de acces;
3. vizualizarea tuturor diagnosticelor asociate unui pacient sub forma unui istoric medical;
4. adăugarea de diagnostice și generarea de radiografii anonimizate pe baza imagisticii încărcate.

1.1. Importanța temei

În contextul aplicațiilor moderne, datorită avansurilor tehnologice, volumul de date generat de către rețelele sociale, rețelele de senzori, Internetul-ului, aplicațiilor cu rol medical, crește cu o rată accelerată și poartă denumirea de Big Data.

Cu toate că Big Data ar putea fi folosit eficient la înțelegerea mai bună a proceselor care ne înconjoară și la inovarea într-o mare varietate de domenii ale activității umane, cantitatea uriașă de date a crescut potențialul de apariție a breșelor de securitate și implicit, la încălcarea intimității datelor personale în anumite circumstanțe, precum: combinarea informațiilor de natură privată cu seturi de date externe ce poate duce la deducția de informații ale altor utilizatori (inferență), cerințe de conformare legală și la reglementări, colectarea și utilizarea de date în scopul de a aduce valoare unei afaceri sau a unui competitor, depozitarea și procesarea datelor cu caracter sensibil într-o locație și/sau mediu nesigure.

Această incidență ridicată de utilizare frauduloasă a datelor a rezultat într-o mulțime de reglementări cu privire la protecția confidențialității datelor, printre care se enumeră GDPR, Patriot Act al SUA sau HIPAA, care au scopul de a defini conceptul de date personale și diferite tehnici de anonimizare a datelor.

1.2. Prezentare generală

Aplicația este compusă din trei mari componente a căror funcționare este interdependentă. O caracteristică importantă a acestora, o reprezintă modularitatea care permite modificări sau schimbări ulterioare asupra părților individuale fără a afecta funcționalitatea aplicației ca un întreg. Comunicarea modulelor se realizează sub forma unui pipeline, astfel că fiecare proces își are intrarea conectată la ieșirea unui alt modul.

Componenta de stocare a datelor este proiectată sub forma unui smart contract ce permite interogarea și persistarea datelor pe rețeaua de test a Ethereum, în urma efectuării unor tranzacții care respectă anumite condiții prestabilite.

Componenta de anonimizareeste responsabilă cu efectuarea inferenței asupra datelor primite de la interfața Web într-o manieră criptată și care elimină caracterul identificabil al datelor. Această componentă este realizată sub forma unei rețele de învățare federată, care permite conectarea unor modele antrenate local, la nivel de clinică sau spital și agregarea acestora într-o manieră criptată pe un server central.

Componenta de colectare și vizualizare a datelor esteimplementată sub forma unei platforme web ce pune la dispoziția doctorilor, posibilitatea de introducere a unui diagnostic sub formă de imagistică medicală, cât și capabilitatea de vizualizare a istoricului medical.

2. NOȚIUNI TEORETICE

2.1 . Date cu caracter personal

În vederea soluționării problemei enunțate de titlul acestei lucrări, și anume, anonimizarea datelor medicale, este necesară definirea atât a spectrului de identificabilitate, cât și a datelor cu caracter personal. Luând în considerare literatura de specialitate, rezolvarea unei anumite provocări își are, în fiecare articol, ca punct de plecare enunțarea în limbaj formalizat a conceptului în vederea înțelegerii sale, prin intermediul unei definiții. În continuare, voi enunța descrierea problematicii pornind de la formulările altor autori.

Legea privind protecția datelor definește datele personale, drept informații ce conțin referiri la etnia individului, convingerile de natură politică, religioasă, condiția sa fizică și psihică, orientarea, cât și istoricul său juridic.

Comisia europeană prezintă în articolele publicate, faptul că, datele cu caracter personal reprezintă orice informație care se referă la o persoană fizică identificată sau identificabilă (PII). Datele care au fost anonimizate, criptate sau pseudo-anonimizate, dar pot fi utilizate pentru reidentificarea unei persoane, rămân considerate ca fiind date personale și sunt vizate de reglementările în vigoare. Pentru ca datele să fie cu adevărat anonimizate, anonimizarea trebuie să fie ireversibilă.În domeniul medical, informațiile personale de sănătate(PHI) sunt colectate, generate, depozitate și prelucrate de către furnizorii de servicii medicale. Aceste informații pot reprezenta istoricul medical, intervenții anterioare sau diferite afecțiuni de natură fizică sau psihică, care pot conduce la identificarea unui individ.

Legea privind confidențialitatea datelor descrie informațiile cu caracter sensibil drept orice informație sau opinie, inclusiv una care poate fi extrasă în urma prelucrării unui set de date, indiferent de validitatea acesteia sau modul de stocare, care poate fi atribuită unei persoane sau care poate facilita determinarea identității acesteia. Informațiile cu caracter sensibil poate reprezenta orice informație care poate fi asociată cu o persoană vie identificabilă. Poate include corespondență, înregistrări audio sau video, imagini, identificatori alfanumerici, cât și combinații a elementelor enunțate anterior. Criteriile ce trebuie a fi îndeplinite pentru a putea clasifica o informație ca fiind personală sunt următoarele: trebuie să fie despre un individ și este rezonabilă presupunerea că identitatea individului poate fi extrasă de pe urma informației, spre exemplu atunci când este inclus numele persoanei, când persoana poate fi distinsă cu ușurință într-un anumit material, sau când informația este atât de unică încât poate fi atribuită unei singure persoane.

Încorporând toate ideile extrase din documentele anterioare, în vederea respectării condițiilor enunțate în acestea și pentru a satisface criteriile enumerate, se ajunge la următoarea definiție: „*Datele cu caracter personal sunt informații care indiferent de modul în care sunt comunicate, stocate și emise, pot fi folosite direct sau indirect în vederea identificării unei anumite persoane*”.

Descrierea stării setului de date este influențată de modul în care o anumită autoritate definește datele cu caracter personal, precum și de reglementările și interpretarea acestora.

În continuare, vor fi prezentate trei forme sub care se pot regăsi datele la nivelul unui serviciu/aplicație, conform definiției anterioare.

1. Identificat– termen folosit atunci când există informație ce permite identificarea directă, precum nume sau adrese. Există și o distincție între conceptul de identificabil(se poate presupune că fie cu datele disponibile, fie în combinație cu alte surse, se poate realiza identificarea) și identificat (identitatea este cunoscută și asociată cu datele).
2. Pseudo-anonimizat– presupune că identificatorii direcți au fost îndepărtați, iar informațiile adiționale necesare reidentificării sunt ținute separat și fac obiectul controlului tehnic și administrativ. Astfel datele pseudo-anonimizate nu ar mai fi identificate, ci identificabile.
3. Anonimizat– procesul de înlăturare a identificatorilor direcți și indirecți (generalizare, substituție, randomizare) pentru un model de partajare a datelor, aducând asigurări rezonabile că datele nu sunt identificabile.

2.2. Definirea conceptului de anonimizare

Conceptul de anonimizare constă în procesul ireversibil de înlăturare a dimensiunii personale a datelor sensibile cu păstrarea formatului și a tipului de date peste care se suplimenteză cu diverse politici sau procese care să nu mai permită identificarea directă sau indirectă, fie utilizând doar informația individuală sau o colaborare cu orice alt set.

Necesitatea anonimizării datelor poate fi atribuită următorilor factori:

- nevoia de a proteja datele sensibile generate în urma unui serviciu;
- creșterea cazurilor de utilizare în scop malițios a datelor cu caracter personal și problemele de confidențialitate rezultate;
- costul uriaș cauzat afacerii ca urmare a utilizării greșite;
- riscuri provenite din factori operaționali, precum externalizarea sau colaborarea cu partenerii;
- respectarea normelor și cerințelor legale.

În funcție de tehnica de anonimizare, se pot obține secvențe realiste orialeatorii pentru datele mascate sau rezultatul poate fi unul determinist, care să fie identic pentru același set de date.

2.2.1. Tehnici de anonimizare

Tehnicile de anonimizare contribuie la mascarea dimensiunii personale a datelor și pot fi calcule statistice, algoritmi sau metode dezvoltate pentru cazuri specifice și trebuie să asigure faptul că tipul de date va rămâne neschimbat.

Substituția presupune preluarea datelor(sau a unei porțiuni de date) de intrare și înlocuirea acestora fie mereu, fie într-o manieră aleatoare cu un alt set de date. Această metodă este folositoare atunci când se dorește păstrarea tipului și formatului datelor mascate și nu se impune un set de date realist ca rezultat, caz în care se folosește o variație a metodei numită substituție de cuvinte(prin intermediul unui fișier de căutare pe baza căruia se vor realiza înlocuirile).

Amestecarea presupune rearanjarea datelor din aceeași coloană, pe un anumit număr de rânduri. În felul acesta se înlătură asocierea dintre un câmp sensibil și o persoană, fără introducerea de noi seturi de informații și păstrarea tipului de date. Cel mai întâlnit caz de utilizare îl reprezintă tabelele cu număr mare de înregistrări peste care se va face o rearanjare, la nivel de grup de coloane, în mod aleator care să nu permită corelarea informației.

Variația numereloreste ometodă statistică ce constă în generarea unui număr într-un interval impus de utilizator. Dacă numărul rezultat în urma variației nu se află în interiorul intervalului, atunci cel anonimizat va fi una din limitele minime sau maxime. Această tehnică se folosește cu precădere în sistemele de Resurse Umane, pentru mascarea salariului angajaților sau în sistemele bancare, aplicată pe depozitele clienților.

Variația datei diferă de metoda anterioară prin generarea unei date într-un interval de timp specificat.

Supresiavalorilor presupune înlocuirea datelor(sau a unui procent din acestea) dintr-o coloană cu caracter sensibil prin valori de NULL/orice alt caracter.

Una dintre cele mai populare metode o reprezintă mascareaîntregului input prin repetarea unui caracter special precum X,*,etc. , folosită îndeosebi la numerele cardurilor de credit sau a numerelor de telefon.

În cazul algoritmilor criptografici, aceștia sunt preferați atunci când considerentele principale sunt reprezentate de integritatea și confidențialitatea datelor, în detrimentul unui set de date realist. Schemele aflate în practică se împart în 3 categorii:

- criptare cu cheie simetrică(utilizarea unei singure chei atât pentru anonimizare, cât și pentru reidentificare) care se poate realiza prin intermediul unui cifru de nivel bloc(DES, 3DES, AES) sau unui cifru de tip stream(RC4) care urmărește generarea unei secvențe de octeți de dimensiunea datelor, pornind de la o cheie dată ca seed;
- scheme cu chei publice precum RSA;

- algoritmi de hash(familia de funcții SHA-3), ale căror principale proprietăți sunt ireversibilitatea, rezistența la coliziuni și output determinist.

2.2.2. Confidențialitatea diferențială

Dintre multiplele pattern-uri și definiții ale anonimizării, cele cu gradul cel mai mare de aplicabilitate în domeniu, al căror formalism matematic conferă garanții de siguranță și care vor fi abordate în cazul lucrării sunt l-diversitatea și confidențialitatea diferențială.

k-anonimatul– termen standard în domeniu folosit pentru a descrie o tehnică de ascundere a identității persoanelor dintr-un grup de persoane similare, unde k reprezintă mărimea grupului. Se poate spune despre un set de date că a obținut k-anonimatul, dacă pentru orice intrare din tabel, există cel puțin (k-1) intrări cu aceleași proprietăți. Această proprietate este realizată în practică prin generalizare și mascare. Generalizarea presupune eliminarea unei părți din date sau înlocuirea cu o valoare comună care să nu mai permită corelarea cu intrări specifice din baza de date.

l-diversitatea– reprezintă o extensie a modelului de k-anonimat care aduce o îmbunătățire a confidențialității datelor personale, prin reducerea granularității reprezentării acestora. Dacă toate persoanele dintr-un set de date au în comun aceeași valoare a unui atribut sensibil, pot fi dezvăluite informații sensibile prin simpla cunoaștere a faptului că aceste persoane fac parte din setul de date respectiv. Astfel o definiție formală ar fi că o clasă de echivalență prezintă l-diversitate dacă are cel puțin un număr l de atribute sensibile reprezentate corespunzător(valori distincte). Un tabel are l-diversitate când orice clasă de echivalență pe care o conține are l-diversitate.

t-apropierea– presupune o îmbunătățire a l-diversității, impunând ca distribuția unui atribut cu caracter sensibil în orice clasă de echivalență este aproximativ egală cu distribuția atributului în întregul tabel(diferența dintre cele două să nu fie mai mare decât un prag t).

Noțiunea de confidențialitate diferențială descrie o promisiune făcută de un deținător de date sau curator, unei persoane vizate, anume: “Nu veți fi afectat, în mod negativ sau în alt mod, permițând ca datele dumneavoastră să fie utilizate în orice studiu sau analiză, indiferent de rezultatul altor studii, prezența altor seturi de date sau surse de informații disponibile”.

Aceasta se bazează pe conceptul de k-anonimat și pe un set de 2 parametrii(ε,δ), care reprezintă:

- cantitatea de zgomot care va fi adăugată;
- pragul care va fi folosit pentru a renunța la categorii cu un grad de specificitate foarte ridicat.

Confidențialitatea diferențială descrie o metodă de adăugare de zgomot matematic la date. Cu confidențialitatea diferențială este dificil să ne dăm seama

dacă o anumită persoană face parte dintr-un set de date, deoarece rezultatul unui algoritm dat va părea în esență același, indiferent dacă informațiile despre o persoană sunt incluse sau omise.

Astfel, dacă un posibil adversar cunoaște informații despre toate persoanele dintr-un set de date, mai puțin una, atunci rezultatul algoritmului de anonimizare nu ar trebui să pună la dispoziție un volum suficient de informații adiționale despre individul rămas pentru a putea fi identificat. Alternativ, dacă se consideră că o persoană furnizează date personale greșite unui colector de date, atunci rezultatul algoritmului nu ar trebui să fie foarte diferit de cel obținut plecând de la setul corect de date.

2.2.3. Date sintetice

O altă abordare pentru anonimizarea unui set de date o reprezintă generarea unui dataset independent care să nu fie derivat din cel original. Această metodă poartă denumirea de generarea de date sintetice, care să nu reproducă identificatori direcți/indirecți și se bazează pe crearea unor intrări care să păstreze din proprietățile și distribuțiile statistice inițiale.

În practică se identifică trei tipuri de date sintetice:

1. date complet sintetice– toate intrările, care identifică direct și indirect, sunt generate sintetic pornind de la un model. Prin urmare, modelul utilizat pentru a produce date sintetice ar trebui să surprindă doar o parte din structura datelor originale. Trebuie realizată o echilibrare între cât de precis este modelul și cât de aproape sunt datele sintetice de datele originale. Este un compromis clasic între utilitatea datelor și confidențialitatea datelor. În urma unei implementări corespunzătoare, datele păstrează suficiente proprietăți statistice ale datelor originale și sunt neidentificabile;
2. date parțial sintetice– doar unele date sunt generate dintr-un model, în mod tradițional datele considerate confidențiale, restul păstrează valoarea inițială. Astfel, chiar dacă apar identificatori în datele rezultate, informațiile deduse ar trebui sa fie minime întrucât restul informațiilor asociate ar fi sintetice;
3. date sintetice hibride– unele intrări sunt generate pe baza unui model și amestecate cu intrări din setul original.

2.3.Tehnologia blockchain

Tehnologia blockchain pune la dispoziție o bază de date distribuită capabilă să stocheze înregistrări și informații publice privind tranzacțiile întreprinse și partajate între toate părțile participante. Blockchain-ul poate fi interpretat sub

forma unui registru public, în care fiecare intrare este persistată în urma realizării mecanismului de consens între toți participanții rețelei.

O primă implementare a acestei tehnologii, care a exemplificat atât nivelul înalt de securitate, cât și potențialul de eficientizare și facilitare a tuturor proceselor ce necesită o instituție centralizată care să reprezinte factorul decident într-un anumit proces, a fost conceptualizată în 2008 într-un articol științific, sub denumirea de Blockchain, publicat de un grup de programatori sub numele de Satoshi Nakamoto. Această publicație propunea o versiune descentralizată de valută care să permită efectuarea plăților între participanți, fără a fi nevoie de o instituție financiară care să medieze transferul.

Proprietate	**Blockchain public**	**Blockchain privat**
Determinare consens	Toți participanții	O organizație
Transparență	Publică	Publică/restricționată
Imutabilitate	Imposibil din punct de vedere computațional de modificat	Posibilitatea de manipulare a datelor
Eficiență	Mică	Mare
Centralizare	Nu	Da
Participare consens	Fără permisiune	Cu permisiune

Tabel 2.1 Compararea tipurilor de blockchain

Caractesticile care conferă specificitate și individualizează tehnlogia blockchain sunt următoarele:

- transparența, toate operațiile care se realizează în cadrul rețelei fiind publice, în așa fel încât validitatea tuturor tranzacțiilor putând fi verificată de orice participant. De asemenea, această proprietate este subliniată și de faptul că nu sunt permise operații de ștergere sau modificare, datele fiind doar concatenate;
- imutabilitatea, realizată atât prin utilizarea funcțiilor criptografice de hash, cât și prin intermediul mecanismului de consens ce presupune obținerea de validări asupra datelor ce urmează a fi persistate de la participanții rețelei;
- anonimat, întrucât toate interacțiunile cu blockchain-ul se realizează cu ajutorul unor adrese generate, care nu divulgă identitatea utilizatorului.

2.3.1. Arhitectura

Arhitectura acestei baze de date distribuite, poate fi reprezentată sub forma unei liste simplu înlănțuite de blocuri, legătura dintre acestea fiind realizată prin intermediul unui hash.

Fiecare dintre aceste blocuri reprezintă o structură de date ce conține o secvență de date, un hash pe conținutul blocului, cât și hash-ul blocului precedent. O funcție de hash reprezintă o metodă de calcul care poate mapa o cantitate

nedeterminată de date într-una de lungime fixă. Această metodă utilizează funcții matematice neinversabile al căror rezultat este ușor de procesat pentru obținerea unei valori hash, dar care fac reproducerea intrării foarte dificilă.

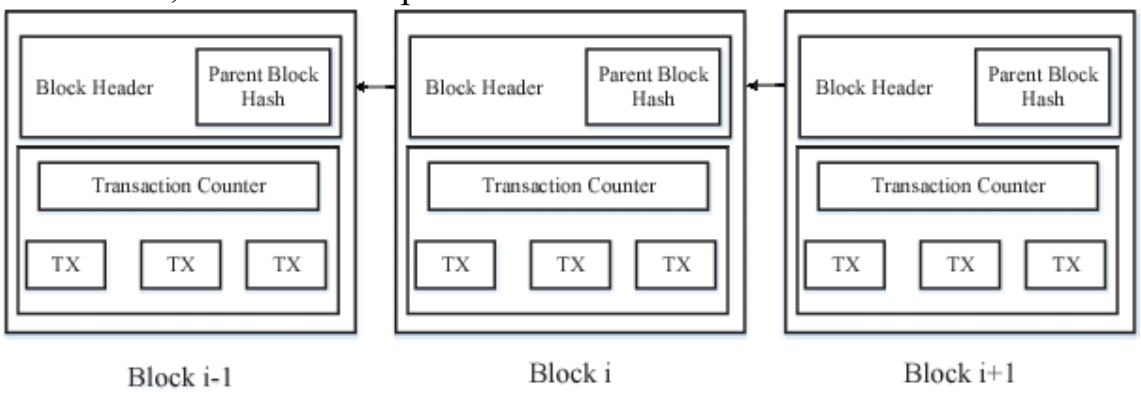

Fig.2.1 Arhitectura unui blockchain

Un hash identifică blocul și tot conținutul său și este întotdeauna unic, așa cum este o amprentă. Odată ce un bloc este creat, hash-ul acestuia este calculat. Schimbarea ulterioară a unui element din interiorul blocului va face ca hash-ul să se schimbe. Cu alte cuvinte, hash-urile sunt foarte utile în detecția modificărilor aduse asupra anumitor blocuri, întrucât modificarea va putea fi semnalizată de celelalte blocuri adiacente.

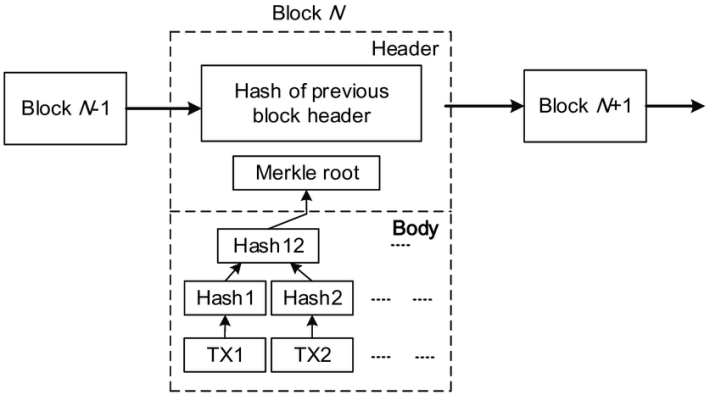

Fig.2.2 Componentele unui bloc

Datele reținute la nivelul unui bloc depind de specificitatea aplicației ce implementează tehnologia blockchain, dar la un nivel general, pot fi descrise următoarele componente:

- versiunea blocului indică politicile de validare pe care blocul curent trebuie să le urmeze;
- rădăcina aborelui Merkle ce conține hash-ul tuturor tranzacțiilor înscrise în cadrul blocului curent;
- timestamp care reprezintă timpul curent la care a fost creat blocul, reprezentat în secunde;

- dificultatea ce indică dimensiunea în număr de biți pe care trebuie să o aibă valoarea hash-ului pentru a putea fi considerat valid. Cu cât numărul de biți este mai mic, cu atât dificultatea de găsire a unui hash care să fie valid, este mai mare;
- nonce, reprezintă o variabilă ce este incrementată, în vederea obținerii hash-ului care să respecte mecanismul de consens implementat la nivelul aplicației;
- valoarea hash a blocului precedent.

Pe lângă elementele descrise anterior care împreună reprezintă antetul blocului, acesta conține și lista de tranzacții confirmate și un contor al acestora. Numărul maxim de tranzacții ce se pot regăsi în interiorul unui bloc depinde de dimensiunea fiecărei tranzacții. Pentru validarea autenticității fiecărei tranzacții, este folosită criptografia asimetrică.

Fiecărei adrese din cadrul rețelei, îi sunt associate o cheie publică și una privată cu ajutorul cărora se realizează mecanismul de autentificare. Prin intermediul cheii private, a cărei valoare trebuie să rămână confidențială, sunt semnate tranzacțiile ce urmează a fi persistate și au un caracter public. Validarea tranzacțiilor se realizează verificând semnătura asociată și suficiența soldului celui care o inițiază.

2.3.2. Mecanismul de consens

Mecanismele de consens (cunoscute și ca protocoale de consens sau algoritmi de consens) permit sistemelor distribuite să lucreze împreună pentru asigurarea siguranței și securității datelor.

Aceste modalitățisunt folosite pentru a valida autenticitatea tranzacțiilor și pentru a menține securitatea blockchain-ului asociat. Acest sistem asigură faptul că toate tranzacțiile legitime sunt înregistrate pe blockchain și că fiecare copie a registrului conține toate tranzacțiile valide.

Protocoalele de consens și algoritmii de consens sunt adesea folosiți în mod interschimbabil. Cu toate acestea, protocoalele și algoritmii sunt diferiți. Un protocol este un set de reguli definite într-un standard care guvernează modul în care un sistem și numeroasele sale părți funcționale conlucrează și interacționează.

De zeci de ani, aceste mecanisme au fost folosite pentru a stabili un consens între nodurile bazei de date, serverele de aplicații și alte infrastructuri ale întreprinderii. În ultimii ani, au fost inventate noi mecanisme de consens pentru a permite sistemelor economice bazate pe monede digitale, să cadă de acord asupra stării rețelei.

Nodurile sau clienții care validează noile tranzacții poartă denumirea de mineri. Mecanismul de consens asigură că toți minerii sunt de acord cu următorul bloc ce urmează a fi persistat și distribuie informațiile din fiecare bloc nou tuturor celorlalți mineri. Oricine poate descărca o copie a blockchain-ului pe dispozitivul său și participa activ la rețea sub forma unui nod.

În funcție de domeniul vizat al aplicației, al numărului de utilizatori, al datelor folosite cât și a structurii care pune la dispoziție rețeaua descentralizată, pot fi identificate mai multe tipuri de mecanisme de consens:

- proof-of-Work, în care minerii concurează între ei pentru calcululvalorii hash necesare validării unui bloc, în schimbul unei recompense. Această modalitate implică un consum foarte mare de electricitate în vederea folosirii resurselor computaționale necesare calculelor, dar furnizează și un nivel de încredere foarte ridicat;
- proof-of-Stake, în care procesul de validare a unui bloc este acordat celor ce dețin cel mai mare volum de monede din rețeaua respectivă. Astfel, timpul de persistare a unei tranzacții este mult mai rapid și persoanele ce dețin miza cea mai mare, sunt recompensate pentru participarea continuă;
- proof-of-Authority, folosită în principal de organizații private, în care asigurările se bazează pe reputație și autoritate, mai degrabă decât pe un consens public.

2.3.3. Contracte inteligente

Contractele inteligente reprezintă seturi de logică computațională și proceduri de tipul scenariu și răspuns, menite să faciliteze, să execute și să impună negocierea sau executarea unui acord.

Reprezintă secvențe de cod partajate într-o manieră descentralizată, care sunt implementate la nivelul rețelei. Participanții care semnează un contract trebuie să convină asupra detaliilor contractuale, condițiilor de încălcare, răspunderile asumate în vederea încălcării și asupra surselor de date de validare externe și apoi acesta poate fi reprezentat sub forma unui set de instrucțiuni și publicat sub forma unui contract inteligent care să automatizeze executarea în lipsa semnatarilor.

Caracteristicile definitorii ale unui contract inteligent sunt reprezentate de autonomie, odată lansate și executate nu necesită intervenția inițiatorilor, autosuficiență, prin capacitatea lor de gestiune a resurselor, prin acumularea lor prin furnizarea de servicii sau utilizarea lor în caz de necesitate și descentralizare, întrucât acestea se execută la nivelul nodurilor rețelei.

Scenariile de utilizare ale unui contract inteligent au o mare varietate, cu prevalență fiind întâlnite următoarele:

- tranzacțiile financiare, întrucât un contract inteligent se mulează foarte bine pe modelul de afacere ce implică echitate, strângere de fonduri, sisteme peer-to-peer de împrumut, cât și servicii de asigurări. Forma tradițională a comerțului, care depinde o agenție centrală precum instituții de schimburi, poate fi eficientizată prin caracteristica robustă a contractului, având ca rezultat reducerea costurilor și evitarea proceselor birocratice;
- piețele de predicție, în care prin mecanismul distribuit de consens, pot ajuta la crearea unor instrumente performante de prognoză. De asemenea, ele pot fi utilizate în cadrul sistemelor de pariuri sau ale celor de votare;

- internet of Things, deoarece pot îmbunătăți comunicarea între dispozitive, facilitarea partajării și permit automatizarea fluxurilor de muncă mari consumatoare de timp într-o manieră sigură din punct de vedere criptografic.

2.4. Inteligență artificială

Inteligența artificială reprezintă oramură derivată din cea a informaticii care se ocupă cu cercetarea și proiectarea de sisteme informatice capabile să simuleze comportament inteligent. Reprezintă abilitatea unui calculator sau a unui agent controlat de calculator să preia informații din mediul cu scopul de a obține o rată cât mai mare de reușită a atingerii a unui țel bine definit, într-o manieră algoritmică și matematică care nu trebuie să se supună metodelor biologice de percepere a unui mediu.

De asemenea, acest domeniu, cuprinde subramurile învățării automate (Machine Learning) și ale învățării profunde (Deep Learning), discipline compuse din algoritmi care caută să creeze sisteme specializate în realizarea de predicții sau clasificări pe baza unor date de intrare.

2.5. Învățare automată

Învățarea automată înglobează studiul științific al algoritmilor și modelelor statistice pe care agenții informatici le folosesc cu scopul efectuării unei sarcini specifice fără ca aceasta să fie programată în mod explicit. Această învățare se realizează prin prelucrarea iterativă a datelor asociate unei probleme specifice și prin adaptarea și ajustarea parametrilor interni de procesare, pentru obținerea unui rezultat cât mai exact.

Întrucât rezolvarea unei probleme nu se poate reduce la folosirea unui abordări generale, s-au identificat o varietate de categorii de algoritmi, fiecare creat având în vedere un anumit număr de variabile și o anumită modalitate de adaptare a parametrilor de învățare, astfel:

- învățare supervizată, care presupune obținerea unei funcții care să asocieze unei intrări, o anumită etichetă, pornind de la un set de exemple de astfel de perechi;
- învățare nesupervizată, în care algoritmii trebuie să extragă și să prezinte caracteristicile importante privind structura datelor, fără a avea puse la dispoziție niște etichete asociate intrărilor;
- învățare prin consolidare, ce reprezintă o arie a învățării automate preocupată cu acțiunile pe care le întreprind agenții software, în vederea maximizării unei recompense cumulative.

2.5.1. Prelucrarea datelor

Etapa de preprocesare a datelor constituie unul din pașii inițiali dinprocesul de antrenare a unui model și se manifestă prin transformarea setului de date astfel încât să fie obținut un format ușor de înțeles și calitativ. Această etapă este prezentă în cadrul fiecărui scenariu de antrenare, dar care diferă prin specificitatea sarcinii ce trebuie realizată.

Astfel, în vederea eficientizării atât a rezultatului, cât și a timpului necesar de antrenare, pot fi avute în vedere operații de standardizare, de eliminare a zgomotului cauzat de valori eronate, prin eliminarea numărului de dimensiuni, fără a elimina atributele ce le conferă unicitate, pentru a facilita analiza, stocarea cât și a aduce datele într-o formă structurată.

În domeniul procesării imagisticii medicale, prelucrarea datelor constă în reducerea dimensiunii imaginilor, extinderea intervalului de valori atribuite intensității pixelilor, aplicarea de filtreîn vederea reducerii zgomotului și a detaliilor nedorite ce pot introduce confuzie în cadrul antrenării rețelei neuronale, mascarea adaptivă ce elimină anumite componente prin setarea unui prag pe diferite intervale de valori ale pixelilor.

În ceea ce privește datele sub format tabelar, etapa de preprocesare implică eliminarea instanțelor ce conțin valori lipsă, discretizarea valorilor numerice pentru încadrarea acestora în anumite intervale țintă și metode de selecție a unui subset minim de atribute folositoare în cadrul procesului de clasificare.

2.5.2. Antrenarea modelului predictiv

Alegerea unui model predictiv reprezintă de fapt selectarea unui algoritm specific tipului de problemă ce se dorește a fi rezolvată și trebuie să aibă în vedere factori precum natura atributului țintă, capacitatea computațională disponibilă, mărimea setului de date.

În rezolvarea sarcinilor ale căror date de intrare sunt imagini reprezentate sub diferite formate și care au ca scop prelucrarea acestora, cât și efectuarea de predicții și clasificări, rețelele neuronale convoluționale reprezintă clasa de algoritmi care a produs cele mai bune rezultate, ele reușind să identifice caracteristicile temporale și spațiale ale imaginilor prin aplicarea de filtre, reducând astfel și numărul de parametrii implicați. Alte domenii în care și-au găsit utilitatea sunt cel al procesării de semnale audio, fiind foarte eficiente în prelucrarea electrocardiogramelor și detecției de defecțiuni în cadrul circuitelor electrice, cât și cel al prelucrării limbajului natural, în care fiecare cuvânt este reprezentat sub forma unui vector tridimensional și prin concatenarea acestora se obțin matrici peste care se pot aplica operațiile de convoluție.

În urma efectuării analizei în amănunt și explorării datelor de intrare, cât și înțelegerea problemei ce trebuie rezolvată, se poate realiza selecția tipului de algoritm ce va fi folosit.

Antrenarea modelului constă în procesul iterativ de furnizare a datelor de intrare, ce are în vedere optimizarea parametrilor specifici algoritmului matematic care modelează problematica propusă. Această ajustare se manifestă se realizează, în vederea minimizării erorii și maximizării predicției produse. Manifestarea antrenării se face diferit, în funcție de specificul problemei, cât și a componenței modelului matematic: distribuții statistice și formule probabilistice, rețele neuronale, funcții sau reprezentări arborescente.

2.5.3. Evaluarea modelului

Evaluarea modelului reprezintă utilizarea sau crearea de diferite matrici, pentru stabilirea punctelor forte și slabe, cât și pentru aprecierea performanței și eficienței predicției obținute în urma antrenării. Unul din cei mai importanți indici ai performanței îl reprezintă acuratețea modelului, definite sub forma raportului dintre numărul de predicții corecte și numărul total de predicții.

$$Acurateţe = \frac{Adevărat\ negativ + Adevărat\ pozitiv}{Adevărat\ negativ + Fals\ negativ + Adevărat\ pozitiv + Fals\ pozitiv} \quad (2.1)$$

Pe lângă acest indicator clasic al performanței, se identifică și o serie de alte scoruri, a căror valoare poate avea un nivel de interes mai ridicat, în funcție de comportamentul urmărit, enumerați în continuare:

- rechemarea, care măsoară procentajul de predicții corecte din totalul de predicții atribuit unei clase;

$$Rechemare = \frac{Adevărat\ pozitiv}{Adevărat\ pozitiv + Fals\ Negativ} \quad (2.2)$$

- precizia, care măsoară procentajul de predicții corecte a datelor noi observate;

$$Precizie = \frac{Adevărat\ pozitiv}{Adevărat\ pozitiv + Fals\ pozitiv} \quad (2.3)$$

- matricea de confuzie, care oferă un raport mai detaliat asupra clasificărilor corecte și incorecte repartizate pe clase. Utilitatea acesteia este în înțelegerea distincției între clase, mai exact, de ce valoarea costului de clasificare incorectă diferă;
- scorul F1, acesta fiind media armonică dintre valoarea rechemării și a preciziei.

$$F1 = 2 * \frac{Precizie * Rechemare}{Precizie + Rechemare} \quad (2.4)$$

2.6. Învățare federată

Învățarea federată reprezintă o metodă de învățare distribuită, de antrenare a modelelor de inteligență artificială, cu ajutorul datelor colectate și stocate de către un număr mare de utilizatori. Antrenarea se desfășoară prin intermediul unui proces iterativ supervizat de un server central, care menține un model global. La fiecare iterație, serverul distribuie starea curentă a modelului global tuturor utilizatorilor, care mai apoi o actualizează prin antrenarea acestui model global pe datele locale și creează un model local. Înainte de începerea unei alte iterații, serverul agregă toate aceste modele locale și actualizează parametrii modelului global. În final, noul model global este distribuit tuturor utilizatorilor participanți. Avantajul principal al utilizării acestei metode, îl reprezintă decuplarea etapei de antrenare de nevoia accesului direct folosite în acest proces.

Sarcinile ideale ce pot folosi acest tip de învățare, au următoarele proprietăți:

- antrenarea pe date reale aflate pe dispozitive mobile oferă un avantaj în detrimentul antrenării datelor general găsite în cadrul centrului de colectare a datelor;
- setul de date folosit conține atribute confidențiale sau mărimea acestuia o depășește cu mult pe cea a modelului;
- în cazul învățării supervizate, etichetarea datelor poate fi realizată direct din interacțiunea utilizatorului.

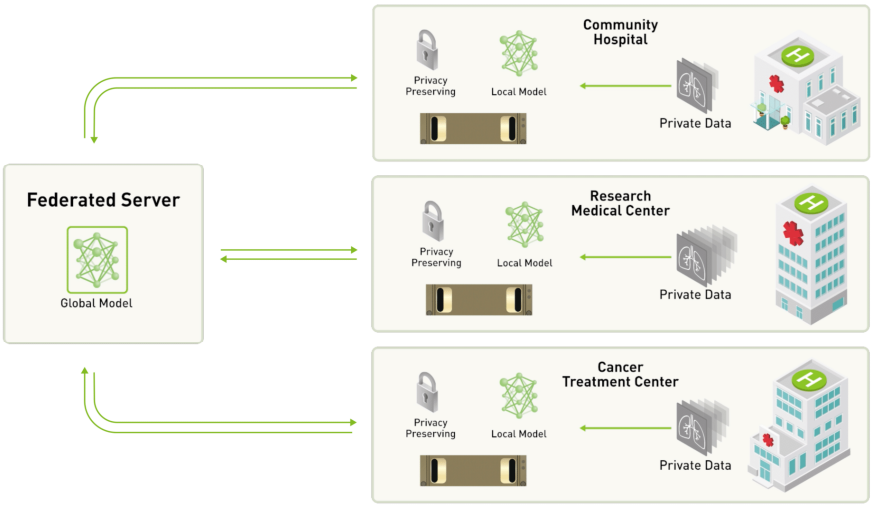

Fig.2.3 Exemplu de rețea de învățare federată

Acest tip de învățare este folosit îndeosebi în domeniul clasificării imaginilor, spre exemplu în prezicerea a căror imagini au o probabilitate mai mare de a fi vizionate de mai multe persoane și în domeniul procesării limbajelor, unde poate fi folosită pentru îmbunătățirea capabilităților de recunoaștere a vocii și de utilizare a mecanismelor de tastare digitală, prin predicții asupra cuvintelor sau frazelor care ar putea urma.

Învățarea federată prezintă avantaje specifice privind menținerea confidențialității comparativ cu antrenarea unor seturi de date persistate. Chiar și stocarea unui set de date anonimizate poate pune identitatea unui individ în pericol, datorită riscului de creare a corelațiilor cu alte seturi de date.

În contrast, informația transmisă pentru procesul de învățare federată este minimul necesar pentru actualizarea unui anumit model, încât puterea de protejare a identității depinde de conținutul acestor informații. De asemenea, întrucât sursa acestor actualizări nu este necesară algoritmului de agregare, acestea pot fi transmise fără meta-date de identificare prin aplicații terțe sau prin rețele de mixare precum Tor.

Arhitectura agentului ce realizează învățarea este formată dintr-un server central și un număr dispozitive client conectate, fiecare din acestea având un set local de date, iar scopul algoritmului de antrenare este găsirea unui punct de minim global pentru o funcție de cost aleasă după specificul problemei, care compară valoarea prezisă cu cea reală. Formalismul matematic ce descrie definiția prezentată anterior are următoarea formă:

$$\underbrace{min}_{w} F(w), unde\ F(w) = \sum_{i \in [N]} \beta_i F_i(w) \tag{2.5}$$

În ecuația anterioară, β_i reprezintă un parametru proporțional cu dimensiunea setului de date local, iar F_i definește funcția de cost folosită local de utilizatorul i.

2.6.1. Protocoale de păstrare a confidențialității

Pentru a introduce noțiunea de computație sigură din punct de vedere criptografic în cadrul unei rețele distribuite trebuie luat în considerare scenariul în care o entitate cu scopuri malițioase controlează un anumit subset de participanți din cadrul rețelei și dorește să intervină asupra procesului de execuție a serviciului. Participanții care se află sub controlul adversarului primesc atributul de corupt și urmăresc instrucțiunile adversarului. Un protocol sigur ar trebui să reziste oricărui atac, în limitele impuse de puterea de control a adversarului. Formalizarea acestui concept și demonstrația siguranței oferită de un astfel de protocol impun enunțarea principalelor proprietăți:

- confidențialitatea descrie faptul că singura informație care poate fi învățată despre datele de intrare ale celorlalți participanți ar trebui să fie cea care se poate deduce din ieșirea propriu-zisă a serviciului;
- corectitudinea ieșirii este garantată pentru fiecare dispozitiv participant;

- independența setului de intrare reprezintă faptul că alegerea setului de intrare a participanților corupți trebuie să se desfășoare fără cunoașterea datelor introduse de participanții onești;
- garanția de primire a ieșirii se asigură că partenerii corupți nu pot preveni ceilalți utilizatori din primirea rezultatului;
- echitatea evidențiază că participanții corupți ar trebui să primească rezultatul ieșirii dacă și numai dacă și restul participanților îl primesc.

Noțiunea care înglobează conceptele definite anterior poartă denumirea de SMPC și permite participanților să realizeze operații computaționale peste seturile lor de date combinate într-o manieră colaborativă, fără a partaja informația folosită între ei.

Unul din protocoalele populare în învățarea federate care pune accentual pe confidențialitatea datelor, este SecAgg care permite serverului central să realizeze agregarea modelelor locale fără a avea acces la fiecare model individual.

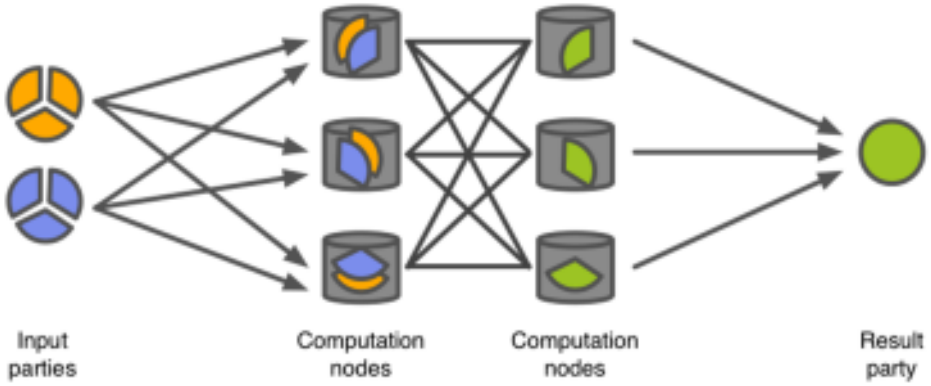

Fig.2.4 Fluxul procesului de SMPC

Acest lucru poate fi realizat printr-un proces numit mascare aditivă, în care fiecare utilizator maschează modelul local folosind o pereche de chei generate aleator, înainte de a-l trimite la server. Generarea cheilor se face în general cu schema Diffie-Hellman, peste care se convine de către oricare doi utilizatori, i și j, la o pereche de valori de inițializare aleatorii$s_{ij}^{(t)}$. Pe lângă aceste valori, unul din cei doi utilizatori creează și o valoare privată de inițializare$s_i^{(t)}$ pentru a proteja confidențialitatea acestuia.

Folosind perechea de valori generate, cât valorile private, utilizatorul realizează mascarea modelului local aplicând următoarea formulă:

$$x_i^{(t)} = w_i^{(t)} + PRG\left(s_i^{(t)}\right) + \sum_{j:i<j} PRG(s_{ij}^{(t)}) - \sum_{j:i>j} PRG(s_{ij}^{(t)}) \qquad (2.6)$$

Protocolul este realizat astfel încât valorile generate random să fie distribuite între utilizatori folosind o schemă de partajare a secretelor și acestea să nu conțină

nicio informație folositoare de sine stătătoare, ci doar suma tuturor contribuțiilor să reprezinte valoarea dorită.

Coruperea a n-1 participanți din cei n conectați în cadrul rețelei este insuficientă pentru a dezvălui informația privată a celorlalte sisteme.

2.7. Container

Un container reprezintă o componentă software care înglobează atât codul necesar, cât și dependințele realizării unei sarcini într-o modalitate ce permite rularea aplicației într-un mod eficient și livrarea facilă și sigură dintr-un mediu într-altul. Principiul de funcționare al containerelor se bazează pe utilizarea mecanismului de virtualizare care se folosește de resursele și apelurile de sistem puse la dispoziție de sistemul de operare gazdă.

Flexibilitatea pe care o aduce procedeul de containerizare este înglobarea tuturor componentelor necesare pentru executarea și obținerea funcționalității complete a unui serviciu. Pe lângă dimensiunile reduse și facilitarea procesului de creare a unei infrastructuri de microservicii, containerele beneficiază și de mecanisme de redundanță care să asigure stabilitate și consistență procesului de interacțiune al utilizatorului.

Containerele permit modularizarea unei aplicații de dimensiuni mai mari și separarea logicii între componente și utilitatea pe care o au în prezenta lucrare este aceea de a simula dispozitivele și interacțiunea dintre acestea în cadrul unei rețele de învățare federate, descrisă în primul capitol al lucrării.

3. STADIUL ACTUAL

3.1.Soluții similare

În contextul utilizării actuale a datelor, există o varietate de implementări a fluxului de anonimizare în cadrul aplicațiilor, printre care se enumeră OpenEMR și soluțiile utilizate de Google, Facebook sau Apple în cadrul serviciilor de îmbunătățire a interacțiunii cu utilizatorul.

Prezentul document are ca scop crearea unei infrastructuri de anonimizare aimagisticii medicale create la nivelul spitalelor și clinicilor și punerea acesteia la dispoziție prin intermediul unei baze de date centralizată accesibilă printr-o platformă Web. Aspectele principale ale acestei interfețe sunt minimalismul și formatul intuitiv de folosire.

Întrucât majoritatea registrelor medicale de specialitate sunt realizate la nivel de spital, accesul la acestea este restricționat doar la câteva persoane, iar colectarea datelor se face fie într-o manieră exhaustivă, doar de către persoane cu cunoștiințe în domeniu, lucru care produce seturi mici de date. Astfel, o deosebire a proiectului față de soluțiile existente este mecanismul transparent de vizualizare a istoricului medical stocat la nivelul blockchain-ului și de obținere a unui rezultat în funcție de intrarea serviciului.

Un alt aspect ce adaugă un plus de valoare este eliminarea insuficiențelor la nivel de seturi de date prin implementarea unei rețele de învățare federate care să creeze un model global folosind o multitudine de seturi locale.

3.1.1. OpenEMR și FHIR

OpenEMR reprezintă cea mai cunoscută arhitectură open-source de management al activităților medicale și de prelucrare în format digital al înregistrărilor pacienților. Întrucât platforma permite utilizarea tuturor formatelor standardizate de agențiile naționale de sănătate și prezintă toate certificările de prelucrare a datelor cu caracter confidențial, aceasta a cunoscut un nivel de adopție foarte ridicat, fiind estimat un număr de 90 de milioane de pacienți deserviți.

Printre carasteristicile importante se enumără posibilitatea de generare de rapoarte standardizate, sistem de gestionare de plăți a costurilor medicale, sistem avansat de realizare a programărilor la consultații, atât din partea doctorilor, cât și pe cea a pacienților și emiterea digitală de rețete la farmacii.

FHIR reprezintă un standard care descrie formatul datelor și elementelor medicale și pune la dispoziție un API intuitiv, specific partajării înregistrărilor digitale din cadrul registrelor medicale.

În cadrul arhitecturii unui registru electronic, aceste două componente sunt complementare și trebuiesc configurate pentru crearea unei interfețe de comunicare cu baza de date care să respecte standardele create în domeniul confidențialității precum GDPR sau HIPAA.

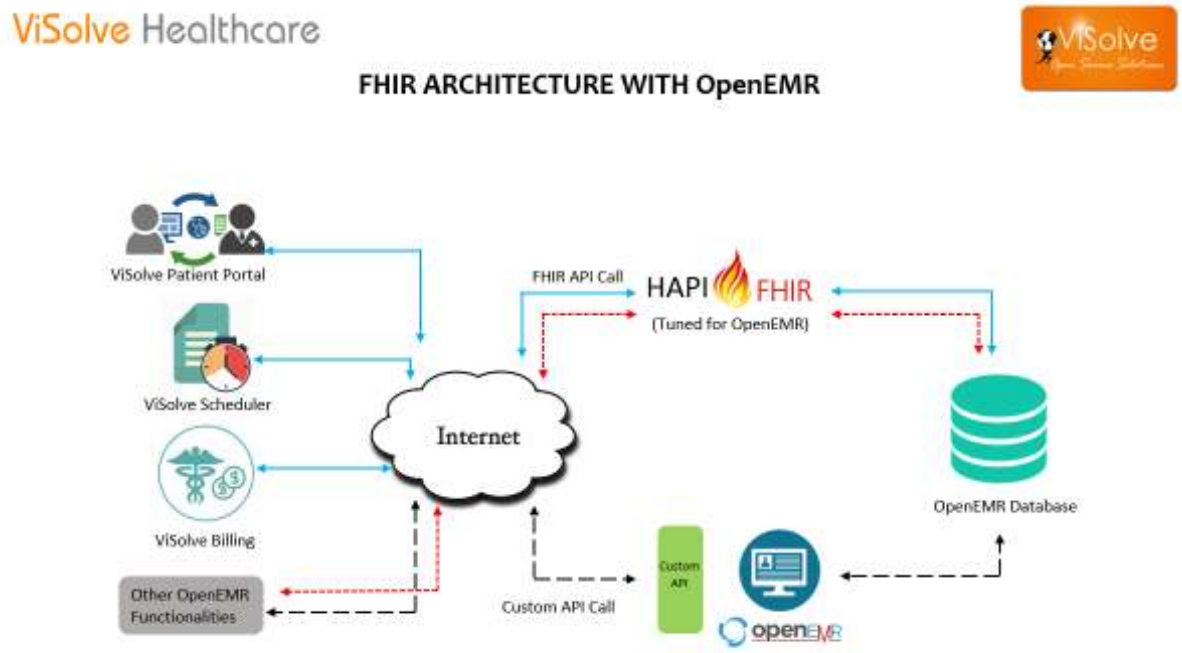

Fig.3.1 Arhitectura unei aplicații ce implementează OpenEMR

3.1.2. Implementări ale fluxului de anonimizare

Mediile de socializare și aplicațiile sau serviciile care se bazează în principal pe interacțiunea numeroșilor utilizatori colectează un volum uriaș de date zilnic. Pentru oferirea unei experiențe cât mai calitative și obținerea de rezultate cât mai specifice utilizatorului, acestea folosesc seturile de date în vederea antrenării unor modele ce învață iterativ din deciziile luate. Cu toate acestea, datele cu caracter personal nu pot fi folosite direct în astfel de modele, astfel că ele trebuie modificate și anonimizate pentru a nu putea fi folosite pentru a identifica un anumit individ.

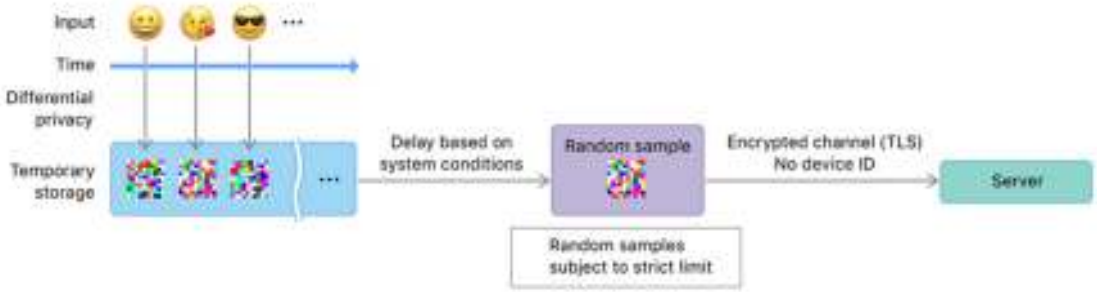

Fig.3.2 Exemplu de aplicare a confidențialității diferențiale

Procesul cel mai des folosit și ușor de implementat îl presupune implementarea unui flux de confidențialitate diferențială. Una din companiile care folosește această modalitate este Apple, care adaugă zgomot la intrările locale obținute de pe dispozitivele dotate cu sistemul de operare iOS, în vederea obținerii de statistici la nivel de populație și augmentarea diferitelor facilități oferite precum funcționalitatea QuickType care generează sugestii pe măsură ce utilizatorul tastează, sugerarea de emoticoane, de titluri de căutare sau de recomandări medicale prin intermediul aplicației integrate de monitorizare a parametrilor.

Un alt exemplu de platformă este un sistem interactiv de interogare care permite directorilor de campanii de marketing să obțină informații statistice asupra interacțiunii utilizatorilor platformei LinkedIn cu conținutul publicat.

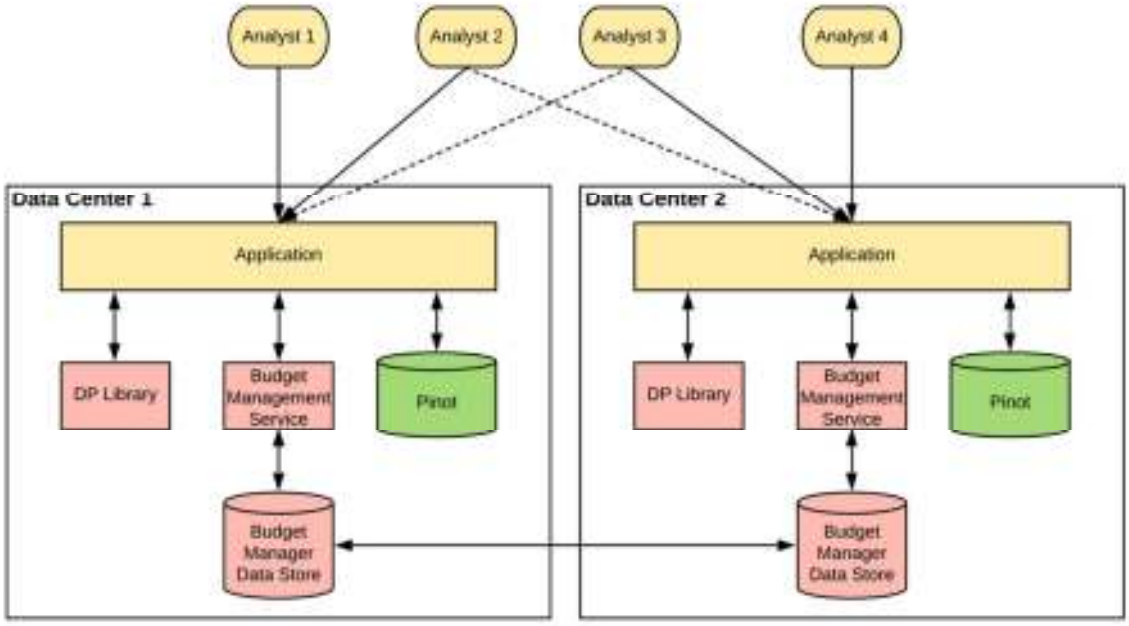

Fig.3.3 Fluxul sistemului Audience Engagement API

Arhitectura aplicației se bazează pe conceptul de analist, fiecare având atribuit un anumit buget de confidențialitate în care trebuie să se încadreze cu interogările dorite. Acest buget de confidențialitate presupune parametrii algoritmului de confidențialitate diferențială care ajustează zgomotul produs asupra datelor și setarea pragului de înlăturare a datelor cu un grad de specificitate ridicat pentru interogarea respectivă.

3.2. Structura sistemului

Sistemul propus este structurat în 3 componente modulare: componenta pentru stocarea datelor, componenta de antrenare a modelului și anonimizare a datelor și componenta destinată vizualizării și interacțiunii cu utilizatorul pusă la dispoziție prin intermediul unei platforme Web pentru introducerea datelor fiecărui diagnostic.

Componenta de stocare a datelor este reprezentată sub forma unui contract inteligent ce prevede mecanisme de gestionare a tranzacțiilor în funcție de nivelul de privilegii ale celui pe care le inițiază. Contractul principal prevede abstractizarea întregului registru electronic digital și pune la dispoziție mecanisme de creare a unui contract secundar menit să fie folosit în gestiunea registrului fiecărui pacient individual.

Pentru folosirea contractului este necesară compilarea acestuia folosind un instrument special, acesta fiind scris în limbajul Solidity, și emiterea lui pe o rețea publică sau privată de blockchain, în cazul de față fiind folosită rețeaua de test Rinkeby.

Componenta pentru realizarea fluxului de anonimizare are la bază o arhitectură de containere docker administrate de un server central care va conține modelul global folosit pentru inferență și antrenare locală. Această arhitectură este bazată pe o structură ierarhică de roluri, fiecare având anumite privilegii și acces la anumite servicii, în ordinea următoare:

- rolul de deținător de date, având capacitatea de a publica un anumit set de date și conferi acces asupra lui;
- rolul de cercetător este atribuit persoanei ce dorește să realizeze antrenarea unui model în cadrul rețelei federate și necesită accesul la un anumit set de date;
- rolul de administrator este îndeplinit de persoana care se ocupă cu crearea utilizatorilor și mentenanța platformei.

Realizarea anonimizării propriu-zise are în componența sa următoarele etape de procesare:

- împărțirea setului de date într-o manieră aleatoare între nodurile participante, încât să existe o distribuție cât mai uniformă a claselor;
- pornirea containerelor ce simulează dispozitivele rețelei de învățare;
- antrenarea modelului folosind datele locale și rețele neuronale convoluționale;
- utilizarea confidențialității diferențiale pentru a masca și elimina posibilitatea de identificare a unui individ;
- validarea setului de date și verificarea acestuia;
- agregarea securizată a modelelor locale folosind protocoale de SMPC pentru obținerea unui model global;
- salvarea modelului global la nivelul fiecărui nod participant.

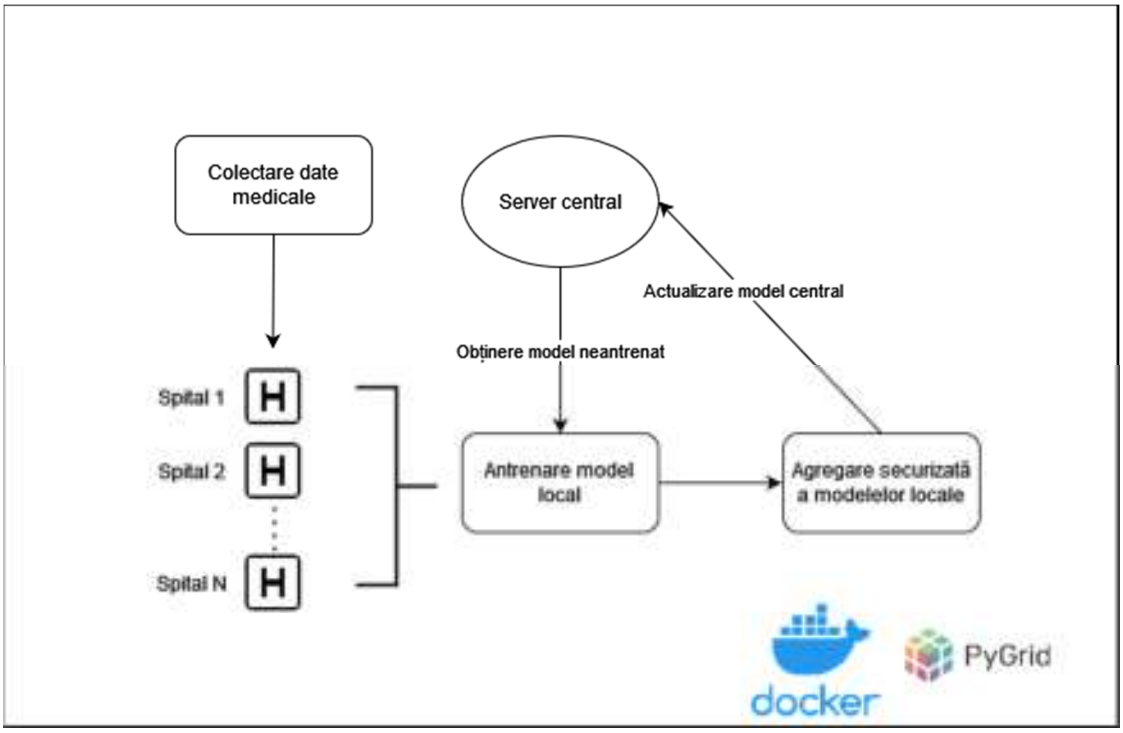

Fig.3.4 Componenta de anonimizare

Componenta care realizează interfațarea serviciului de anonimizare cu utilizatoruleste pusă la dispoziție prin intermediul unei platforme Web pentru introducerea de diagnostice și vizualizarea istoricului medical. Întrucât aceasta realizează medierea între componenta de colectare și serviciul de anonimizare, este compusă, la rândul ei, din mai multe servicii:

- serviciul de server web care realizează comunicarea cu celelalte componente;
- serviciul de server side rendering, care permite vizualizarea eficientă a informațiilor de pe blockchain prin servirea de pagini statice clientului;
- serviciul de stocare a informațiilor în cadrul contractului inteligent sub formă de tranzacții;
- Serviciul de REST API folosit pentru comunicarea cu rețeaua federată de învățare și realizarea inferenței într-o manieră sigură.

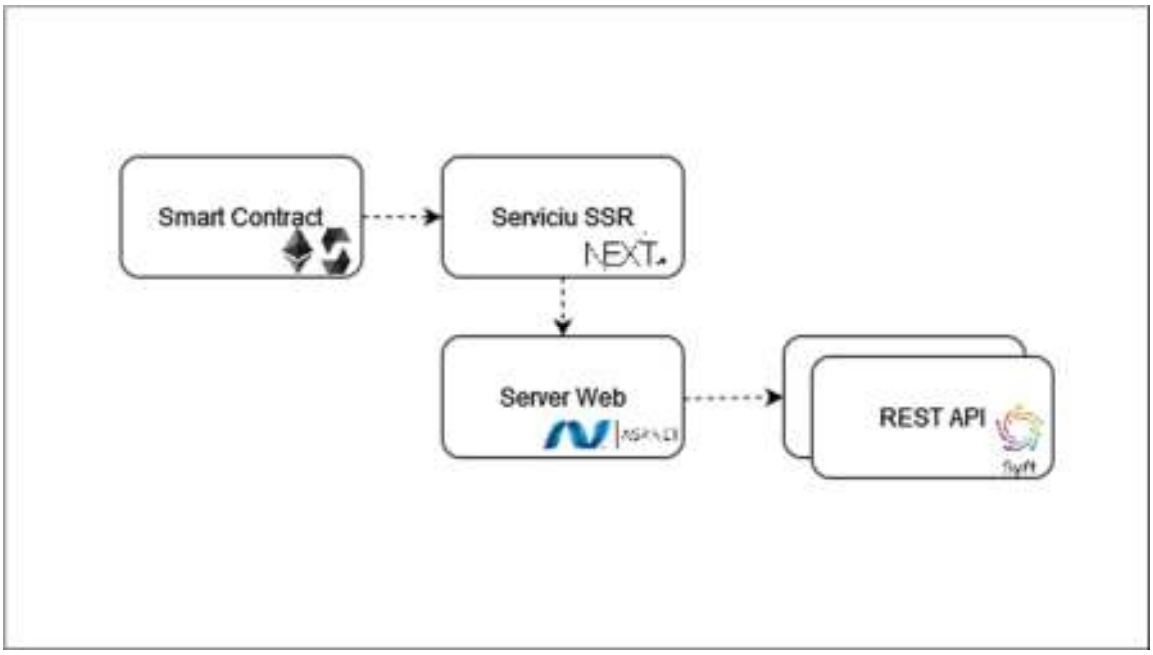

Fig.3.5 Componenta pentru interacțiunea cu utilizatorul

4. IMPLEMENTARE

Modul de realizare a aplicației a implicat simularea locală a rețelei de învățare federată prin intermediul instrumentelor puse la dispoziție de containerele docker și folosirea interfeței Swagger și Postman pentru dezvoltarea serverului web, urmând ca aplicația să fie transferată pe un serviciu public de găzduire în cloud.

4.1. API-uri și biblioteci utilizate

4.1.1. Node.js

Node reprezintă un mediu open source și cross-platform de rulare a codului Javascript pe partea de server, folosit pentru realizarea de aplicații scalabile ce urmează să fie expuse în rețea. Performanța se datorează atât mecanismului intern de gestionare a evenimentelor, cât și pe utilizarea motorului de execuție V8 preluat din cadrul browser-ului Google Chrome și adaptarea acestuia.

O aplicație bazată pe această arhitectură rulează într-un singur proces, fără a mai crea un nou fir de execuție pentru fiecare cerere, model ce implică o coadă non-blocantă de evenimente de scriere și citire. Modalitatea prin care face acest lucru este punerea la dispoziție a unui set de primitive asincrone pentru operații de I/O. Atunci când codul necesită efectuarea unei operații precum citirea de informații din rețea, accesarea unei baze de date, Node.js nu blochează firul de execuție, irosind cicluri de procesor, ci reia operațiile în momentul de întoarcere al răspunsului.

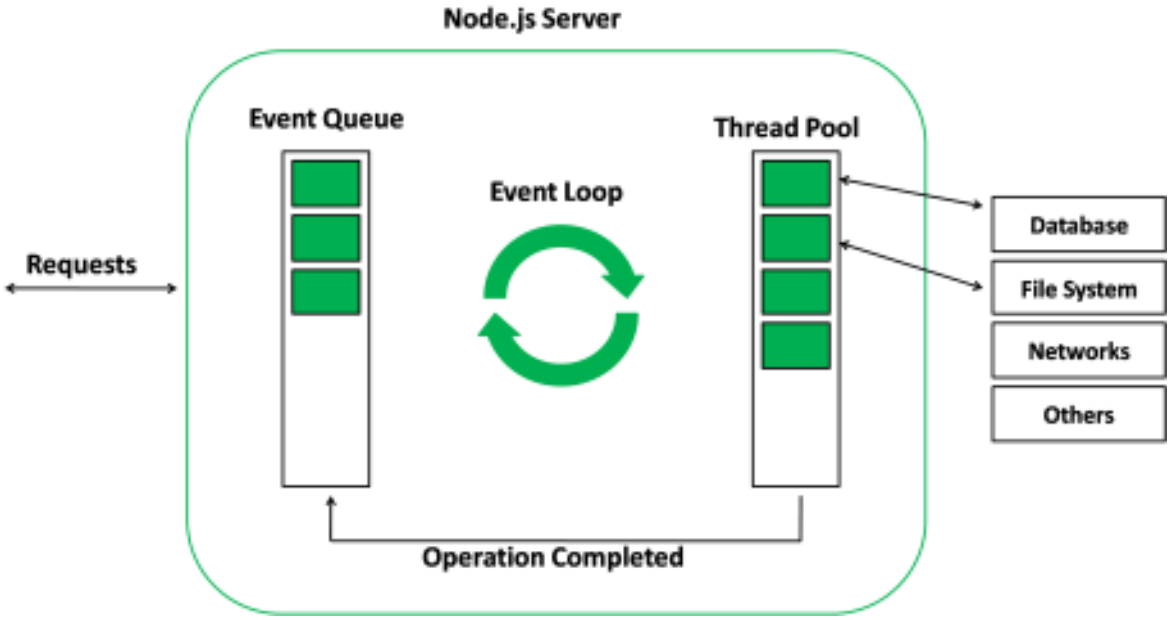

Fig. 4.1 Coada de evenimente

Node.js preia una din operațiile care se află în coada de evenimente și începe execuția acesteia în fundal. În momentul în care operația a fost finalizată, kernel-ul sistemului de operare transmite acest lucrul aplicației și în cadrul cozii va fi adăugat apelul corespunzător de tratarea a răspunsului.

4.1.2. React

React reprezintă o librărie menită sa faciliteze procesul de creare a unei interfețe pentru interacțiunea cu utilizatorul folosind o structură arborescentă de module numite componente.

O astfel de componentă reprezintă un mix de cod HTML și cod Javascript care încapsulează toată logica necesară afișării unei părți din cadrul interfeței. Toate aceste componente, reprezintă părți reutilizabile de cod, care pot fi combinate pentru obținerea unui ansamblu mai complex.

Pentru a permite dezvoltatorilor îmbinarea elementelor celor două limbaje, a fost realizată o extensie a sintaxei limbajului Javascript, numită JSX. Componentele vor fi scrise fie sub forma unei funcții sau a unei clase ce moștenește o primitivă ce conține modelul unei componente și va fi afișată prin intermediul unei funcții de randare, care în loc să recreeze toate elementele definite, va prelua doar modificările aduse de la ultima stare cunoscută și va încerca să aplice doar modificările necesare pentru afișarea noii structuri.

În cadrul librăriei React, pentru fiecare obiect DOM, există o corespondență cu un obiect virtual, sub forma unei copii de mici dimensiuni, ce poartă denumirea de VDOM. Întrucât manipularea elementelor grafice reprezentate de acest DOM este lentă, React generează un arbore de elemente în memorie, echivalent cu DOM-ul real. Modificarea acestui arbore virtual este mult mai eficientă, întrucât nu presupune și afișarea elementelor, ci doar actualizarea componentelor lor prin intermediul unei interfețe conceptualizate.

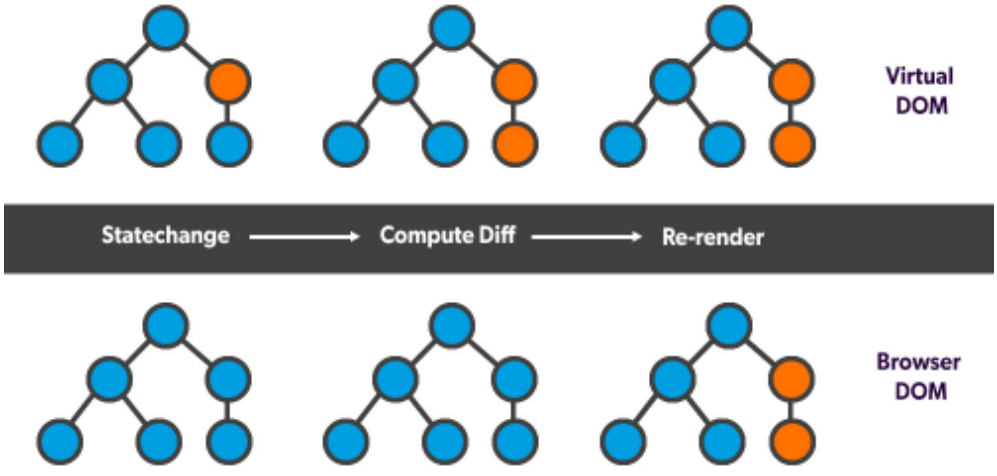

Fig. 4.2 Manipularea Virtual DOM-ului

4.1.3. NextJS

NextJS reprezintă un framework open-source, construit peste React, dezvoltat de compania Vercel, care pune la dispoziție toate componentele necesare realizării unei aplicații Web ce respectă standardele contemporane. Principalele facilități pe care le pune la dispoziția utilizatorului sunt acelea de realizare a interfeței grafice, gestionare a rutelor de navigare între diferitele părți ale aplicației, posibilitatea de randare a conținutului dinamic și static, cât și mecanisme ce conferă scalabilitate, performanță și o bună experiență dezvoltatorului.

Componentele folosite în cadrul lucrării și care valorifică cel mai bine utilitatea acestui kit de dezvoltare de aplicații Web sunt următoarele:

- Routing;
- SSR;
- SSG;
- Capabilitatățile puse la dispoziție pentru preluarea datelor din medii externe de stocare.

Routing-ul reprezintă modalitatea de gestionare a accesului utilizatorului la diferitele pagini componente ale aplicației și este capacitatea de navigare în cadrul interfeței. În cadrul unei aplicații Next, rutarea se face prin intermediul sistemului de fișier, având ca punct de plecare directorul *pages*, tratând fiecare fișier scris în limbajul Javascript sau o extensie a acestuia, sub forma unei rute.

Pentru a facilita navigarea și îmbunătăți performanța, Next preîncarcă fiecare pagină existentă și navigarea între acestea se realizează utilizând o primitivă definită în cadrul framework-ului. Această primitivă permite atât rutarea statică, generată din structura imbricată de directoare din sistemul de fișiere, cât și posibilitatea de rutare dinamică prin utilizarea unei sintaxe specifice în numele unui fișier, acceptând parametrii din interacțiunea paginii cu utilizatorul.

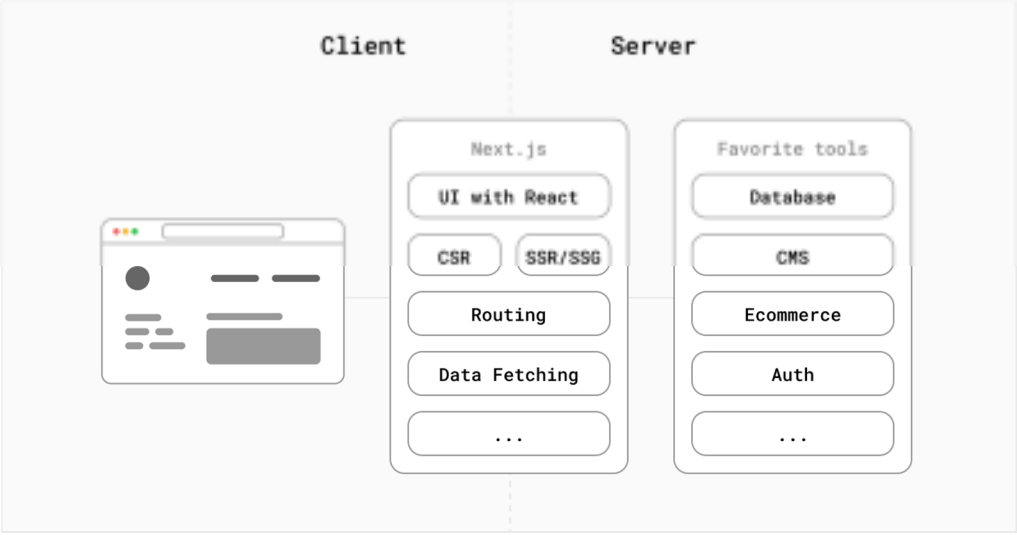

Fig. 4.3 Arhitectura unei aplicații Next.js

Prin componenta de Server-Side Rendering, codul HTML al fiecărei pagini este generat pe server pentru fiecare cerere generată de utilizator. Acest lucru este preferat în cazul situației în care sunt prezentate date dinamice actualizate frecvent. DOM-ul generat, datele în format JSON, cât și instrucțiunile în cod Javascript care conferă interactivitate paginii, sunt mai apoi servite clientului. Astfel, vizualizarea componentelor statice se realizează foarte rapid pentru utilizator, iar React se folosește de restul datelor pentru a oferi componente interactive, precum elemente de gestiune a unui eveniment produs de utilizator.

În simplificarea procesului de preluare a datelor, Next oferă posibilitatea de a obține conținutul dorit prin apeluri asincrone și încărcarea acestuia în cadrul unei componente, înainte de încărcarea sa și servirea către utilizator.

Modulul de Static Site Generation, permite încărcarea paginii la momentul compilării, lucru care permite încărcarea întregului conținut pus la dispoziție de o aplicație Web la prima rulare. Codul sursă conținut se va ocupa de gestiunea interacțiunii, dar ideal, va produce puține modificări asupra interfeței.

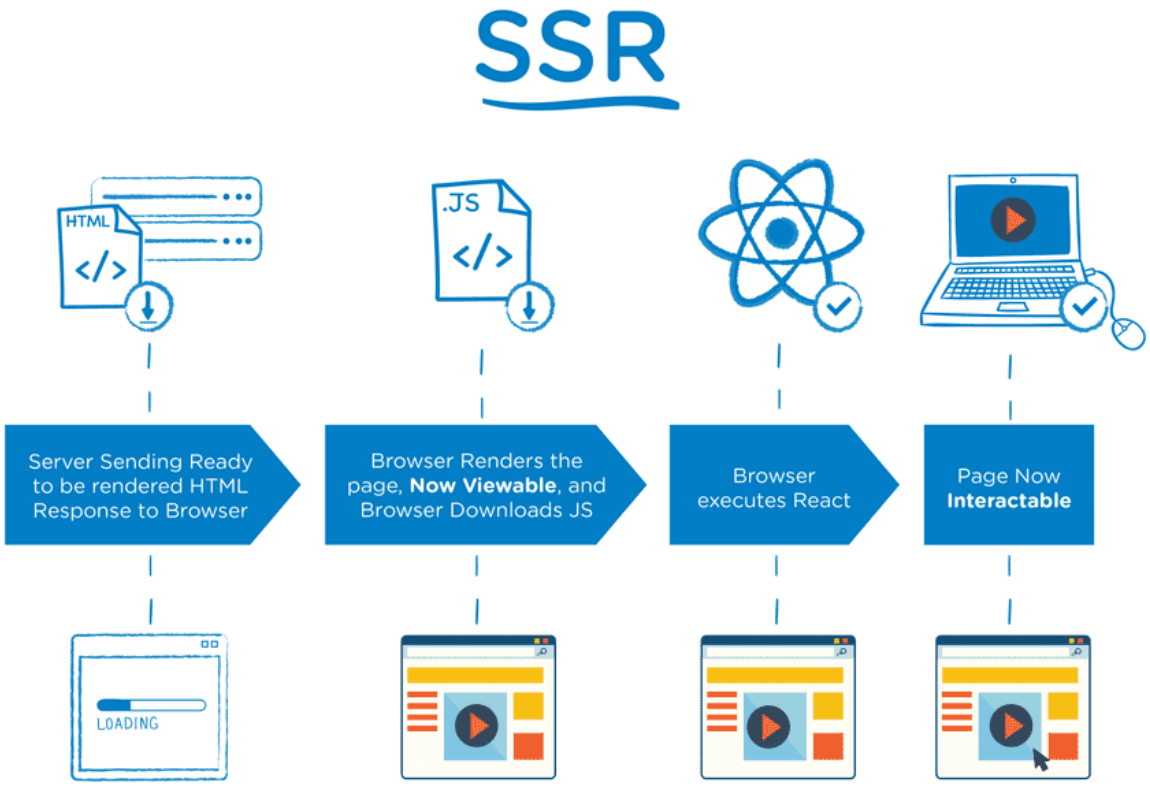

Fig. 4.4 Explicarea conceptului SSR

4.1.4. Web3.js

Web3 reprezintă o librărie care permite dezvoltatorilor de aplicații să interacționeze cu rețeaua de blockchain numită Ethereum. Resursele digitale, precum criptomonedele sau contractele inteligente reprezintă componenta centrală a aplicațiilor descentralizate. Cu toate acestea, pentru a realiza operații asupra acestor componente emise pe rețeaua de noduri, este nevoie de realizarea de tranzacții.

În cadrul Ethereum, nodurile pun la dispoziție utilizatorilor interfețe de nivel jos pentru înaintarea de tranzacții, numite JSON RPC expuse prin diferite opțiune precum protocolul HTTP sau socket-uri. Această interfață este un format de codificare al textului ce permit proceselor să primească date.

Librăria Web3 realizeză o extensie a limbajului Javascript ce permite comunicarea cu această interfață și astfel oferă posibilitatea de conectare a aplicației web la rețeaua Ethereum prin HTTP, rețea găzduită fie pe un nod local de test sau de un furnizor ce operează asupra unor puncte de acces în blockchain.

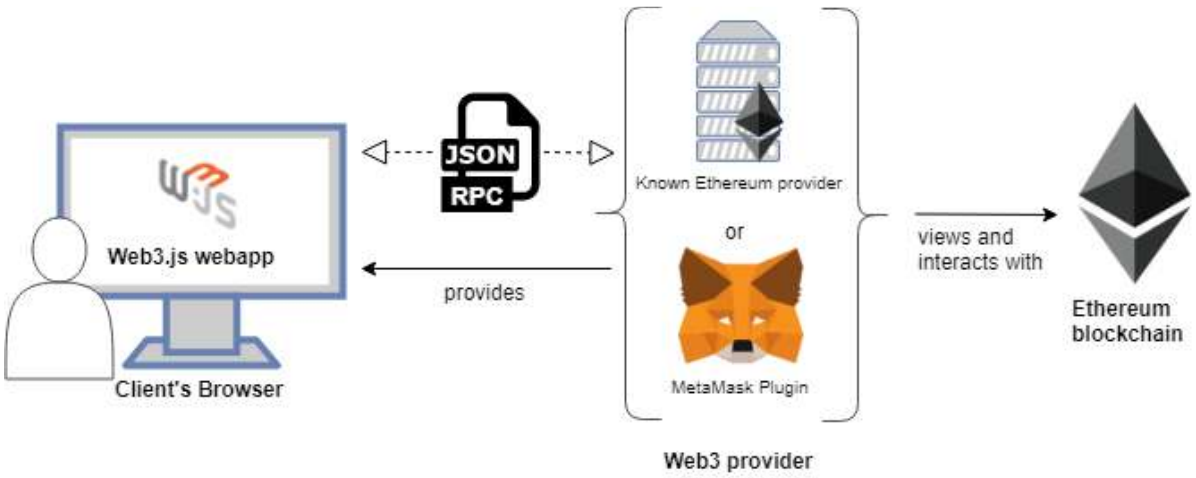

Fig. 4.5 Interacțiunea cu aplicații descentralizate

4.1.5. Metamask

Metamask reprezintă o aplicație software de tip portofel criptografic care vine sub forma unei extensii de browser și îndeplinește două funcționalități principale:

- permite crearea unei noi instanțe de portofel și stochează cheile private și publică asociate. Cheia private este folosită la realizarea semnăturilor pentru aprobarea tranzacțiilor emise de initiator către blockchain;
- permite conectarea și interacțiunea cu majoritatea aplicațiilor descentralizate.

Crearea unui cont pe acest utilitar se realizează prin intermediul algoritmului BIP39 care generează un set de cuvinte ușor de memorat, pe post de frază de acces, care să permit generarea deterministă de perechi de chei.

Algoritmul presupune generarea unei valori a entropiei, care reprezintă și nivelul de securitate al frazei generate și presupune un interval de 128 – 256 de biți. Pasul următor reprezintă calcul unei sume de control folosind o parte din biții entropiei generate și o funcție de hash precum SHA256. Scrierea binară a entropiei este împărțită în grupuri de câte 11 biți, a căror valoare convertită în reprezentare zecimală va indexa o listă de cuvinte în limba engleză.

În continuare, se poate folosi funcția PBKDF2 cu valoarea de salt formată din cuvântul "mnemonic" și fraza obținută anterior, un număr de iterații setat la valoarea 2048 și funcția pseudo-aleatoare HMAC-SHA512 pentru a deriva perechile de chei folosite în crearea de instanțe ale unui portofel criptografic.

4.1.6. Pytorch, Scikit-Learn, Numpy, Pandas

Bibliotecile enunțate în subtitlu înglobează modulele de bază din kitul de dezvoltare al unui cercetător în domeniul inteligenței artificiale, punând la dispoziție toate uneltele necesare în antrenarea, testarea și evaluarea unui model de rețele neuronale.

Librăria Pytorch este una din cele mai populare și eficiente pentru crearea unui model de învățare profundă într-o manieră flexibilă și modulară, punând la dispoziție numeroase funcționalități în funcție de specificul problemei pentru care este antrenat o anumită rețea.

Scikit-learn pune la dispoziție modalitatea de împărțire a setului de date în partea de antrenare și cea de testare a performanțelor obținute.

Numpy și Pandas reprezintă unelte care facilitează procesul de structurare, manipulare și explorare a datelor pe tot parcursul procesului de proiectare a unui model.

4.1.7. PySyft

PySyft este un framework de învățare profundă, construit peste PyTorch, care fac posibilă executarea procesului de învățare automată într-o manieră sigură și confidențială. Scopul acesteia este de a populariza tehnicile de menținere a confidențialității în domeniul inteligenței artificiale, făcându-le accesibile prin mecanisme familiare și intuitive, puse la dispoziția cercetătorilor.

Platforma își propune implementarea simplă și integrarea facilă și flexibilă a acestor mecanisme în procesele de învățare federată, SMPC și confidențialitate diferențială.

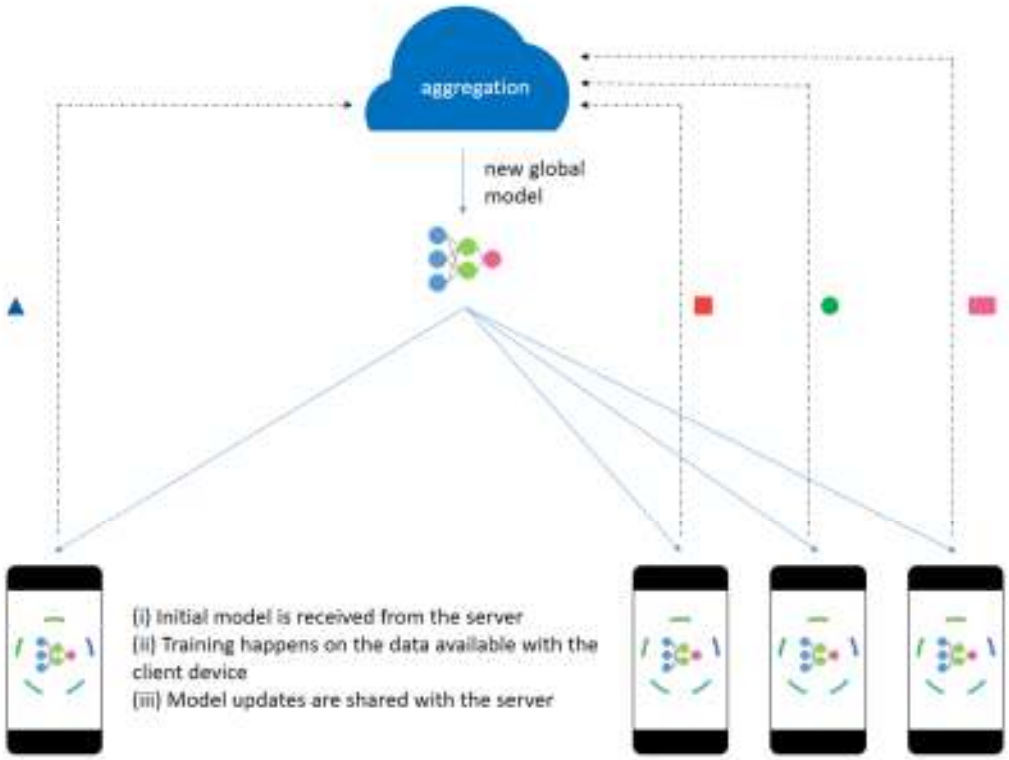

Fig. 4.6 Arhitectura unei rețele de învățare federată

4.2. Cerințe proiect

Cerințele aplicației vor fi enunțate pentru fiecare modul independent, având în vedere componentele arhitecturale ale soluției.

Componenta server, realizată cu ajutorul framework-ului NodeJS, are ca obiectiv implementarea unui sistem de mediere a mediului de stocare a datelor aplicației și utilizarea unui model de inteligență artificială disponibil pe un server centralizat.

Serverul are rolul de a extrage informații introduse în interfața Web, a le prelucra astfel încât să corespundă structurii contractului inteligent în cadrul cărora vor fi stocate. Totalitatea operațiilor cu mediul de stocare vor fi realizate prin intermediul tranzacțiilor și rolul principal al acestei componente este acela de preîncărcare a tuturor informațiilor asociate unui utilizator de pe aplicația descentralizată și de randare a interfeței la nivelul serverului pentru oferirea unei experiențe fluide, care să mascheze timpii mari de încărcare asociați operațiilor asincrone cu blockchain-ul.

Componenta client, implementată prin facilitățile puse la dispoziție de Next.js, are rolul de asigura o experiență intuitivă de navigare și utilizare diferitelor tipuri de utilizatori gestionați de aplicație.

Atât înregistrarea și autentificarea se va realiza prin intermediul unui cont existent în cadrul rețelei de blockchain Ethereum, care cuprinde și facilitatea de autorizare asupra conținutului afișat prin procedeul de semnare cu ajutorul cheii private asociate contului.

Întrucât aplicația este destinată utilizării atât de personalul medical, cât și de cel nespecializat, paginile puse la dispoziție sunt împărțite în funcție de rolul utilizatorului. Doctorii vor putea vizualiza atât pagina de cereri de acces asupra registrului medical unui anumit pacient, cât și pagina cu toți pacienții asupra cărora au drepturi de editare. Aceștia vor avea posibilitateade granularizare a vizualizării, atât la nivel de registru medical individual, cât și de diagnostic, precum și capabilități de adăugare de diagnostice.

Pacienții vor putea vizualiza și interacționa cu schimbările ce au loc în cadrul istoricului medical propriu, într-un format detaliat, cât și posibilitatea de generare a unui raport asupra intrărilor dorite. De asemenea, acești utilizatori vor putea vizualiza totalitatea cererilor primite de la doctori pentru a le accesa datele, precum și capacitatea de confirmare sau refuzare a acestora.

4.3. Arhitectura generală a sistemului

Arhitectura generală a soluției dezvoltate este alcătuită din 3 module principale evidențiate în figura X:

- componenta de stocare a datelor obținute atât din interacțiunea utilizatorilor cu platforma, cât și din interacțiunea cu modelul antrenat, care introduce mecanisme de autorizare și de control al fluxului prin procesul de semnătură digitală;
- componenta de învățare federată, care adună modelele antrenate local în cadrul unui cluster de containere și le agregă la nivelul unui server centralizat într-o manieră sigură din punct de vedere criptografic, astfel încât să nu apară scurgeri de informații;
- platforma web pentru interfațarea funcționalității cu utilizatorii aplicației, care permite vizualizarea datelor și obținerea rezultatului în urma interacțiunii cu modelul antrenat.

Întrucât problematica enunțată este cea a datelor cu caracter personal și dat fiind modul ales de stocare al datelor, sub formă de tranzacții pe blockchain, implementarea componentei Web, cât și a celei de stocare a implicat doar utilizarea a câtorva servicii care să faciliteze comunicarea asincronă necesară persistării și vizualizării datelor. Aceste servicii reprezintă câteva componente software popularizate în industria tehnologică în cadrul acestui domeniu și anume serviciul de server-side rendering și capacitatea de generare a paginilor statice propusă de Next.js, cât și funcționalitățile limbajului Solidity.

În completarea soluțiilor open-source anterior menționate, rezolvarea problemei propuse de lucrare a implicat și utilizarea unor tehnologii dezvoltate cu ajutorul limbajului Python, aici fiind evidențiată importanța ecosistemului PyGrid

al framework-ului PyGrid. Prin intermediul acestuia, a fost posibilă crearea și menținerea rețelei peer-to-peer necesară pentru a reprezenta o rețea de dispozitive implicate în procesul de învățare federate.

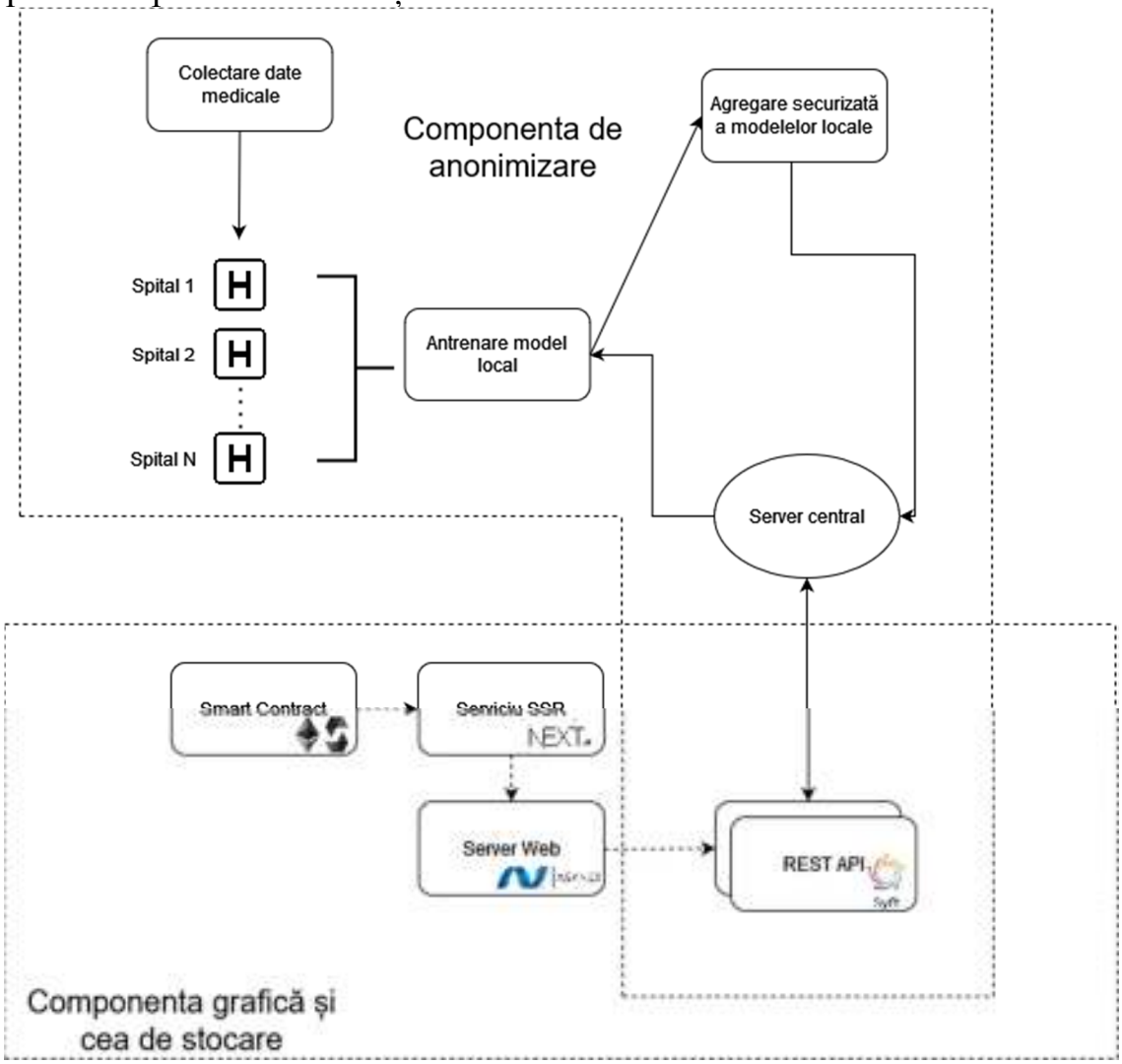

Fig. 4.7 Arhitectura generală a sistemului

De asemenea, soluția software pune la dispoziție și capacități de adăugare a confidențialității diferențiale, a procesului de agregare securizată a parametrilor, cât și de susținere a modelelor antrenate local și agregarea acestora într-un model global partajat global.

4.4. Colectarea datelor

Întrucât problematica temei și normele europene de protecție a datelor nu permit dezvoltarea de procese proprii de colectare și etichetare a datelor, setul de date utilizat pentru modelul propus de anonimizare este unul de radiografii în vederea detectării pediatrice a pneumoniei, revizuit asupra calității imaginii și caracterului reprezentativ de un medic radiolog specialist.

Motivația din spatele alegerii acestui set de date, l-a reprezentat posibilitatea de diagnosticare a afecțiunilor și bolilor uzuale, întrucât, conform Organizației Mondiale a Sănătății, pneumonia este responsabilă de decesul unui număr de 2 milioane de copii cu vârste mai mici de 5 ani anual, fiind considerată cauza principală a mortalității infantile.

Conform unui raport al OMS, aproximativ 95% din cazurile de pneumonie clinică care debutează la copii sunt întâlnite în țările aflate în dezvoltare, astfel că regiunea geografică din care au fost preluate majoritatea datelor este reprezentată de Asia de Sud-Est și Africa.

Elementul principal care dictează diagnosticul de pneumonie, îl reprezintă analiza datelor radiografice, întrucât radiografiile regiunii toracale permit diferențierea între tipurile existente ale pneumoniei. Conform acestei rațiuni, au fost colectate un număr de 5232 de radiografii provenite de la un număr de 5856 de pacienți.

Etichetarea datelor, asemănător procesului de colectare care necesită și aparatură dedicată, implică cunoștințe de specialitate în domeniul medical și s-a realizat prin conturarea cauzelor principale de apariție a pneumoniei, adică tipul patogenilor contractați: bacterii sau virusuri. Astfel, datele existente au fost împărțite în trei clase, normal, bacterial și viral și repartiția a fost realizată prin crearea de directoare specifice în sistemul de fișiere, pentru procesul de antrenare, respectiv de testare și mai apoi, subdirectoare asociate claselor.

Lucrarea ce descrie exhaustiv obținerea și utilitatea setului de date se intitulează “Identifying Medical Diagnoses and TreatableDiseases by Image-Based Deep Learning”. Acest articol științific reprezintă un studiu, în care este descrisă aplicabilitatea mecanismului de învățare prin transfer de la un model antrenat pe setul de date ImageNet și performanța obținută de acesta pentru efectuarea de diagnostic privind prezența pneumoniei la cazurile pediatrice.

Alegerea mecanismului de transfer a parametrilor obținuți de la modelul preantrenat este motivat în primul rând de numărul mic de instanțe prezente în setul de date, întrucât durata antrenării unui model nou inițializat aleator care să obțină o acuratețe satisfăcătoare ar dura câteva săptămâni. În al doilea rând, este luată în considerare și procesul dificil de colectare și preprocesare în volum mare a imaginilor medicale, având în vedere caracterul identificabil al datelor extrase din radiografii.

4.5. Explorarea datelor

Procesul de explorare a datelor are scopul de întelegere a setului de date folosit în cadrul procesului de antrenare, cât și a influenței acestuia asupra comportamentului obținut la final.

În această analiză s-au urmărit mai multe elemente de interes, precum distribuția statistică a datelor pe clasele definite de studiul care s-a ocupat cu colectarea datelor, reprezentarea în diferite formate a radiografiilor în vederea

identificării unui anumit criteriu de diferențiere. În acest sens s-au efectuat calcule privind procentajul de copii sănătoși și cei bolnavi de pneumonie, proporția tipurilor de boală identificate, parametrii care vor afecta rezultate obținute prin procesul de inferenă asupra datelor de testare.

În figura următoare se poate observa distribuția aproximativ egală a imaginilor în primele 2 clase, cu o discrepanță foarte mare la cea de a treia, având în vedere prevalența mult mai mare a agenților bacterieni în țările subdezvoltate care au luat parte la procesul de colectare. De asemenea, între cazurile negative de pneumonie, s-au regăsit și anumite boli sau afecțiuni care au afectat analiza.

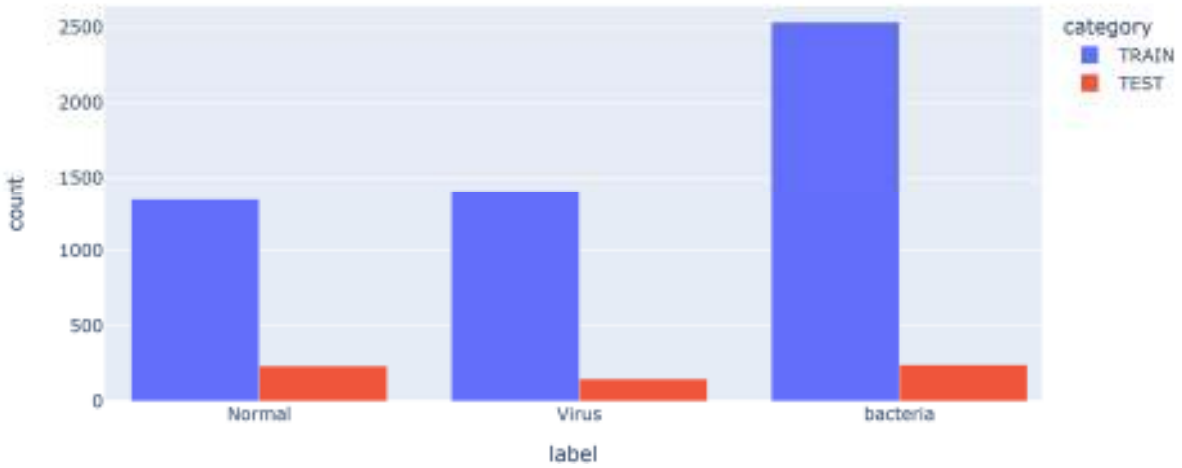

Fig. 4.8 Distribuția radiografiilor pe clase

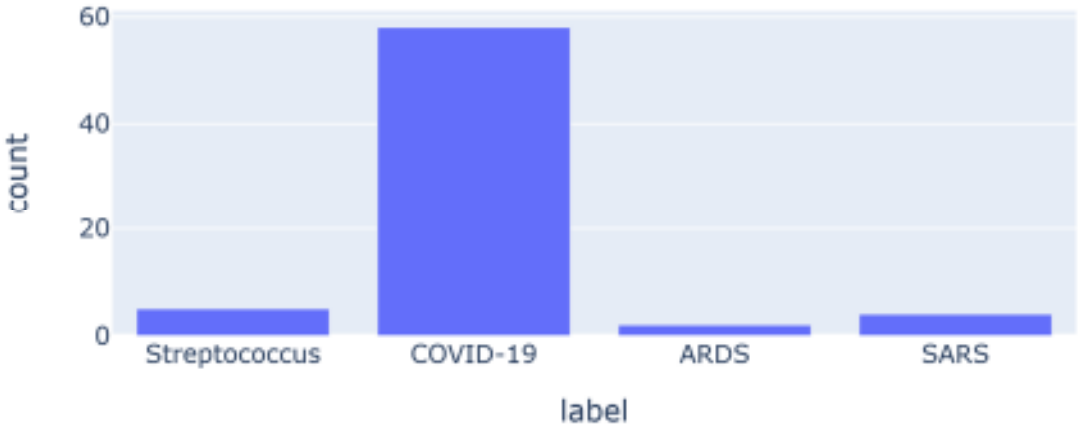

Fig. 4.9 Afecțiuni care au afectat analiza

Din corpusul selectat pentru analiză, se poate constata un raport de 1:4 între copiii sănătoși și cei suferinzi de orice tip de pneumonie.

Pe lângă această constatare, au fost supuse vizualizării și o serie de imagini selectate aleator, câte 2 per clasă. Se poate observa faptul că în cazurile de

pneumonie sunt prezente anumite pete de culoare albă care semnalează infecția, iar personalul de specialitate poate determina pe baza acestora prezența unor abcese sau unei efuziuni de pleură.

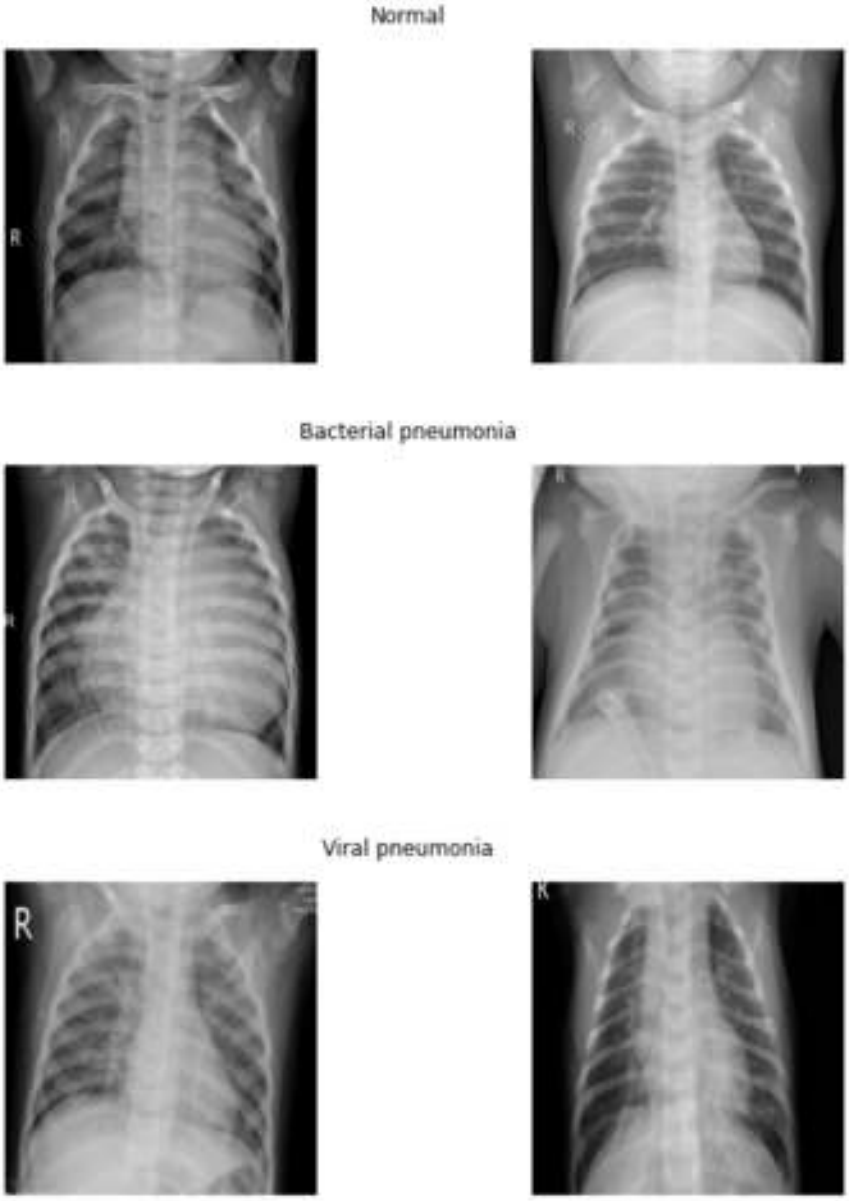

Fig. 4.10 Vizualizarea claselor din setul de date

În continuarea analizei, am folosit și două tehnici specific analizei radiografiilor de pneumonie și anume spectrul de magnitudine, al cărui rol este acela de a evidenția localizarea infecției, cât și mărimea acesteia.

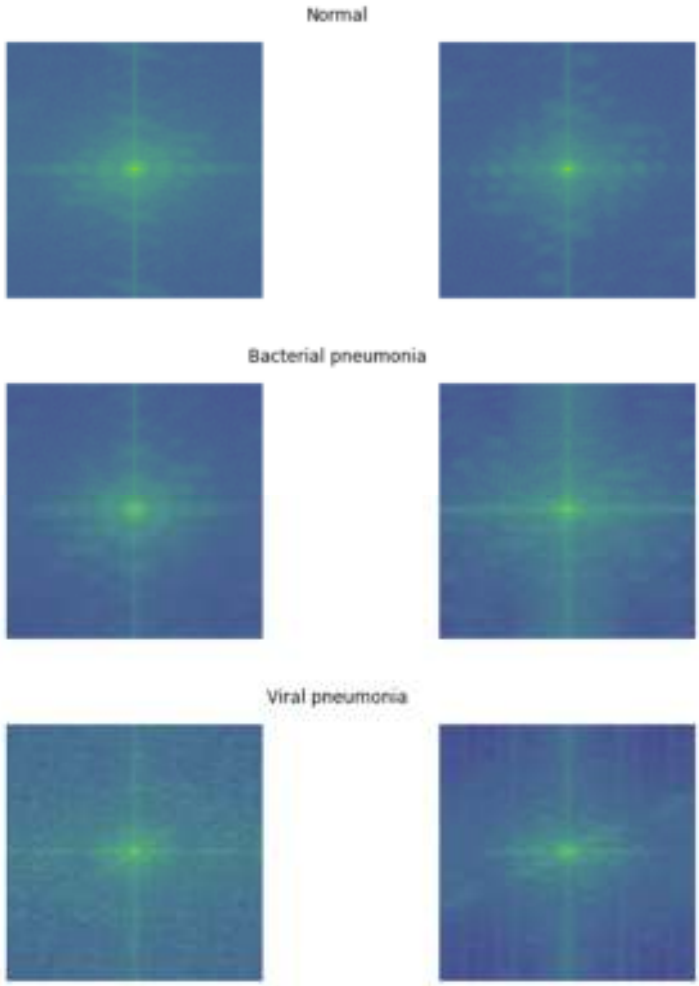

Fig. 4.11 Compararea spectrelor de magnitudine

În final am calculat histogramele tuturor imaginilor și le-am grupat pe clasele asociate. Apoi, în continuare, am dorit să calculez media tuturor valorilor la nivel de clasă, pentru a obține o reprezentare sugestivă a valorii de intensitate a pixelilor. În figura următoare pot fi regăsite numărul de apariții mediu al valorii pixelilor, printr-o serie de 3 grafice suprapuse. Numărul mare de pixeli de 0 este datorat formatului imaginii, cât și a fundalului negru pe care sunt realizate radiografiile. În continuare, poate fi observată și o diferență dintre distribuția de intensitate în cazul unui copil sănătos și cea a unuia bolnav. Astfel, acele pete albe care au putut fi observate și în figura 4.10, pot fi extrase și din această creștere a numărului de pixeli cu intensitate ridicată cu valori între 180-210.

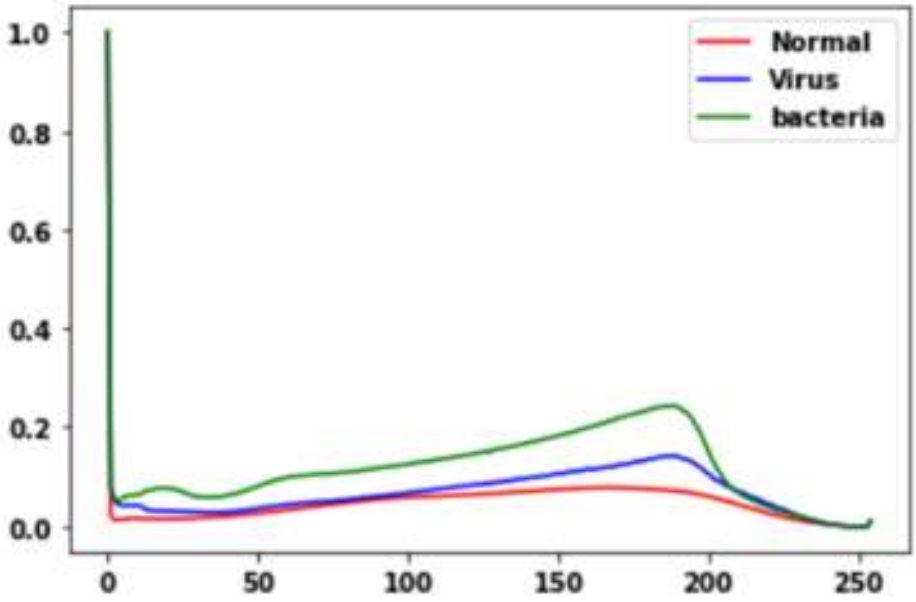

Fig. 4.12 Histogramele asociate celor 3 clase

4.6. Antrenarea modelului

Antrenarea modelului se realizează cu procedee de confidențialitate diferențială implementată cu ajutorul librăriei Opacus, la final obținându-se un model valid care va realiza predicția diagnosticului pentru o radiografie dată ca intrare.

Arhitectura modelului va fi compusă din modelul preantrenat ResNet18 și va conține ca și intrare o rețea de radiografii redimensionate asupra cărora au fost efectuate anumite seturi de transformări geometrice precum rotații, translații, operații de scalare și mărire.

Parametrii ce țin de confidențialitatea diferențială au următoarele semnificații:

- max Grad Norm este distanța maximă în spațiu vectorial, între gradienții imaginilor, înainte de a fi agregați la următorul pas;
- multiplicatorul de zgomot reprezintă cantitatea de zgomot eșantionată și adăugată la rezultatul obținut în urma agregării gradienților dintr-un batch;
- Delta a cărui valoare ar trebui să fie mai mica decât inversul dimensiunii setului de antrenare, este fixat la 0.00001;
- Epsilon reprezintă conexiunea între cantitatea de zgomot, adică implicit uzabilitatea datelor obținute și nivelul de confidențialitate dorit.

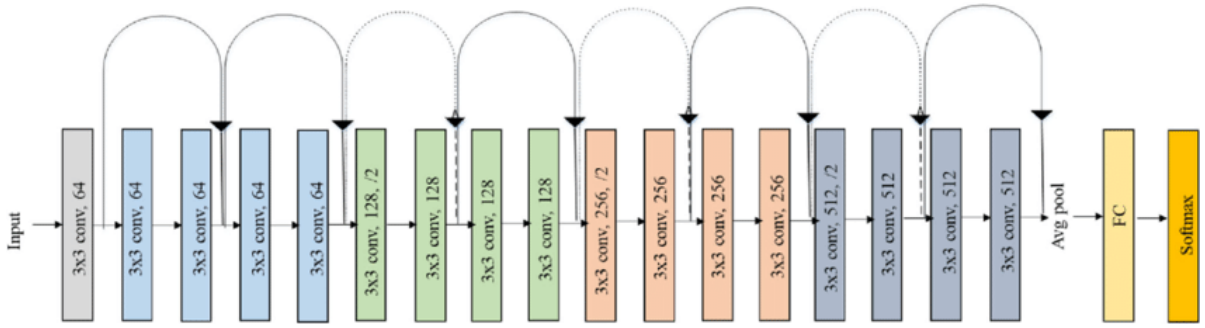

Fig. 4.83 Arhitectura modelului

Configurarea hiperparametrilor specifici antrenării a fost realizată după consultarea mai multor modele menite să trateze problematica abordată, prin combinarea valorilor acestora în funcție de performanță și capacitățile hardware disponibile. În plus, pentru selectarea valorilor folosite în păstrarea confidențialității a fost utilizată literatura de specialitate[X] care tratează mai multe valori, în vederea rezolvării acestui tip de problemă. Valorile folosite pentru configurarea hiperparametrilor sunt reprezentate în tabelul următor:

Hiperparametru	Valoare
Rezoluția imaginii	224
Dimensiune batch antrenare	32
Dimensiune batch validare	32
Funcția de pierdere	Cross Entropy Loss
Numărul de epoci	20
Rata de învățare	0.0001
Optimizator	Adam
Funcție de activare	Softmax
Multiplicator de zgomot	1.3

Tabel 4.9 Tabel hiperparametrii

Resursele hardware care au susținut procesul de antrenare sunt reprezentate de 8 procesoare grafice Tesla K80. Timpul de antrenare pentru o epocă a fost de aproximativ o oră, de unde timpul total necesar antrenării modelului poate fi estimat la aproximativ 20 de ore.

Fig. 4.104 Acuratețea per epocă pentru antrenare și testare

Acest model va fi parte componentă a modulului de anonimizare și va fi folosit în felul următor: doctorul va încărca în momentul creării unui diagnostic o imagistică medicală sub formă de radiografie a zonei toracice, această va fi trecută prin model pentru obținerea unei imagini anonimizate, cât și a unei etichete de clasificare, iar la final poza va fi încărcată pe serviciul de stocare furnizat de Azure Cloud și va fi returnat solicitantului un obiect JSON ce conține link-ul de acces și eticheta rezultată.

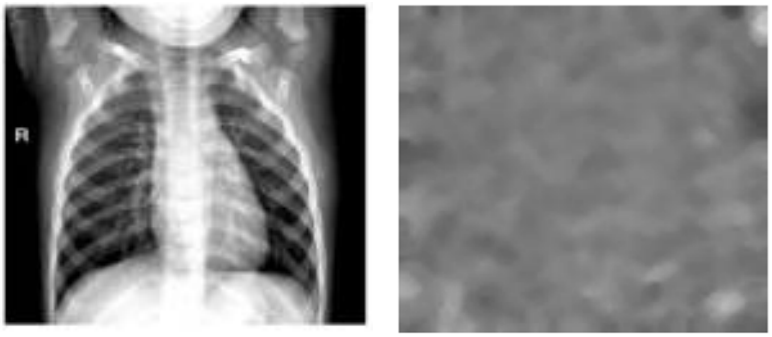

Fig. 4.115 Formatul de intrare și ieșire a unei imagini

4.7. Implementare software

Implementarea serverului a fost realizată cu ajutorul limbajului Javascript, cu ajutorul framework-urilor Next.js versiunea 12.1.6 și NodeJS versiunea 16.14.2. Pachetele software anterior menționate sunt necesare pentru implementarea funcționalității de server-side rendering și reprezintă o aplicație de front-end de sine stătătoare. Serverul pe care este menținută această aplicație este implementat

cu ajutorul limbajului ASP.NET Core. Conectarea celor două componente, pentru a beneficia atât de capacitățile de redirectare și utilizare a protocolului HTTPS, cât și de posibilitate de generare de fișiere statice pentru a putea fi servite clientului se face prin intermediul unui proxy, în fișierul de configurare astfel:

```
builder.Services.AddNextjsStaticHosting();

var app = builder.Build();
app.UseRouting();

app.UseEndpoints(endpoints => {
    endpoints.MapControllerRoute(
        name: "default",
        pattern: "{controller}/{action=Index}/{id?}");

    endpoints.MapNextjsStaticHtmls();
});

app.UseNextjsStaticHosting();

app.Run();
```

Pentru facilitarea dezvoltării aplicației, a fost utilizat utilitarul "create-next-app" care creează ierarhia sistemului de fișiere asociat rutelor disponibile navigării în cadrul aplicației. Interfața grafică a fost creată cu ajutorul framework-ului ReactJS, versiunea 18.2.0 și elementele principale ale componentei asociate clientului are următoarea structură:

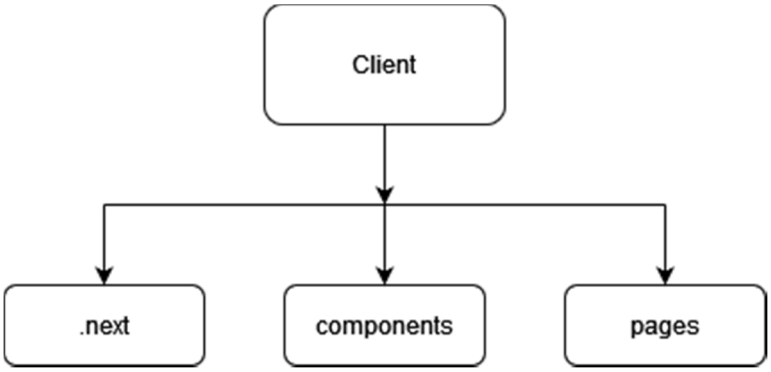

Fig. 4.126 Structura componentei client

Directorul ".next" conține fișierele HTML statice și script-urile de Javascript care conferă interactivitate paginilor generate cu ajutorul funcționalității de static site generation a componentei Next.

Directorul "pages" conține rutele configurate, atât dinamic, cât și static prin structura imbricată a subdirectoarelor și fișierelor conținute. În cadrul acestora din urmă, este inclusă implementarea modalității de autentificare cu aplicația descentralizată susținută de contractul inteligent și preluarea conținutului persistat pe blockchain.

În final, directorul “components” înglobează toate componentele React utilizate în cadrul interfeței Web.

4.7.1. Compilarea și emiterea contractului inteligent

În procesul de scriere al contractului inteligent a fost folosită platforma Remix IDE, care pune la dispoziție o rețea de blockchain local menită să simuleze un caz real de utilizare. Mecanismele pe care platforma le oferă utilizatorului prin intermediul interfeței grafice sunt:

- posiblitatea de selectare a mediului de lucru, fie că este vorba de blockchain-ul local pentru testare și dezvoltare, sau o rețea publică specifică, pentru implementare;
- posibilitatea de interacțiune cu parametrii tranzacției;
- capabilități de debug la nivelul codului și funcțiilor apelate;
- interacțiunea cu contracte existente prin furnizarea de adrese;
- selectarea unei versiuni de compilare a codului Solidity.

În cadrul procesului de dezvoltare a fost folosită versiunea 0.8.9 a compilatorului de Solidity, care aduce anumite modificări și îmbunătățiri privind procesul de compilare și sintaxa disponibilă.

Directorul “ethereum” conține toată logica asociată procesului de emitere a unui contract inteligent.

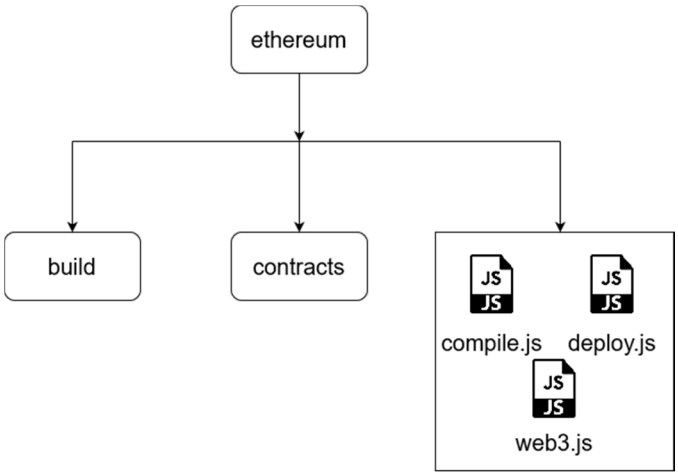

Fig. 4.137 Ierarhia directorului Ethereum

Fișierul “compile.js” utilizează compilatorul de Solidity din librăria “solc”, furnizându-i parametrii necesari privind conținutul, limbajul și configurările ieșirii. Acest fișier va produce crearea directorului build, prin generarea unui fișier în forat JSON care conține atât limbajul mașină asociat contractului, cât și interfața prin care vor putea fi accesate metodele sale.

Directorul “contracts” conține totalitatea fișierlor “.sol” utilizate care conțin codul sursă al contractului, dezvoltat în formatul OOP propus de limbajul Solidity.

Fișierul “deploy.js” realizează emiterea contractului pe rețeaua selectată. Pentru a putea realiza acest lucru este necesar codul compilat la pasul anterior extras din fișierul generat, un furnizor cu tot cu portofelul digital asociat accesat prin fraza de 12 cuvinte obținută la momentul creării. Utilizând un cont dorit din cadrul portofelului, se poate semna o tranzacție de creare de contract digital, având ca parametru codul mașină corespunzător.

În urma acestei operații, în terminal va fi furnizată adresa publică care poate fi folosită pentru instanțierea contractului.

Pentru a facilita efectuarea de operații pe partea de server, au fost realizate două fișiere adiționale care exportă atât o instanță a întregului registru electronic, cât și o metodă de instanțiere a registrelor medicale individuale.

Motivația dezvoltării unui blockchain local a reprezentat-o atât viteza de realizare a tranzacțiilor, aproape instantanee comparativ cu timpul mediu de 10-15 secunde asociat unei rețele publice, cât și posibilitatea de emitere gratuită a contractelor, întrucât orice aplicație descentralizată are un cost asociat în funcție de nivelul de complexitate al operațiilor implementate și de dimensiunea totală a codului.

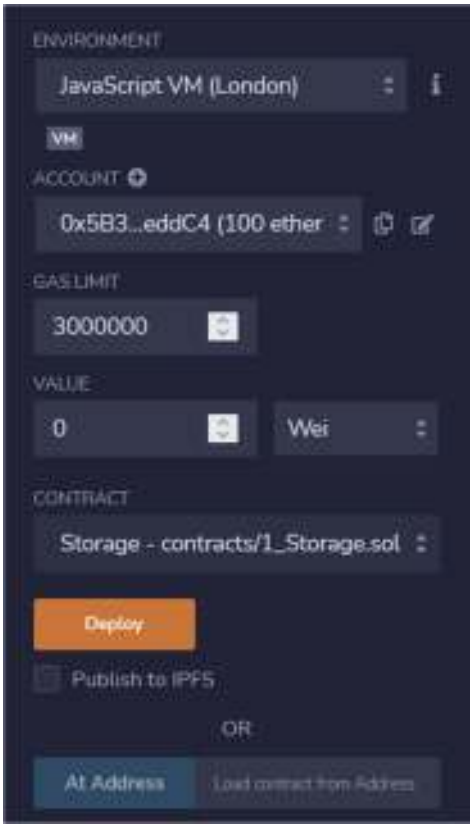

Fig. 4.148 Interfață Remix IDE

4.7.2. Autentificare și autorizare

Primul pas în procesul de autentificare este conectarea contului de pe rețeaua Rinkeby cu un serviciu API, numit Moralis și accesat printr-un pachet pus la

dispoziție de framework-ul Node, ce asigură facilități de organizare a utilizatorilor și permit comunicarea cu aplicațiile descentralizate. Moralis permite gestionarea sesiunii utilizatorului curent și reține adresa acestuia prin care putem garanta accesul în blockchain și realiza tranzacții.

Pentru conectarea cu serviciul Moralis, este necesară semnarea acestei cereri de autentificare cu ajutorul cheii private regăsite în portofelul de la Metamask. Având în vedere acest proces care nu necesită informații precum adrese de e-mail, parole sau alte informații ale utilizatorului, este garantată anonimitatea.

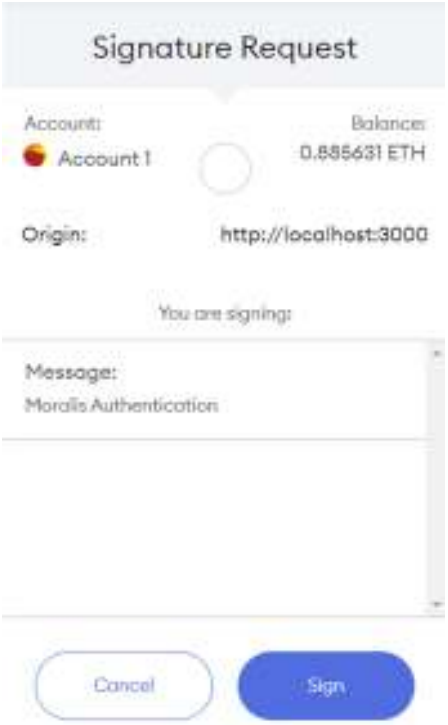

Fig. 4.159 Semnarea cererii de autentificare

În continuare, adresa contului cu care se face autentificarea va fi căutată în baza de date Moralis pentru a verifica dacă există un utilizator asociat. În caz afirmativ, vor fi afișate informațiile contului și meniul de navigare în funcție de rolul pe care îl îndeplinește utilizatorul, acela de doctor sau pacient. În caz contrar, utilizatorul va fi nevoit să introducă câteva informații de identificare personală, cât și biometrică. Fiecare câmp necesar a fi completat este verificat pentru corectitudine folosind expresii regulate pe partea de server. De asemenea, adresa de e-mail și codul numeric personal vor fi verificate pentru unicitate. Nerespectarea unei condiții enunțate anterior, va determina afișarea unui mesaj de eroare specific.

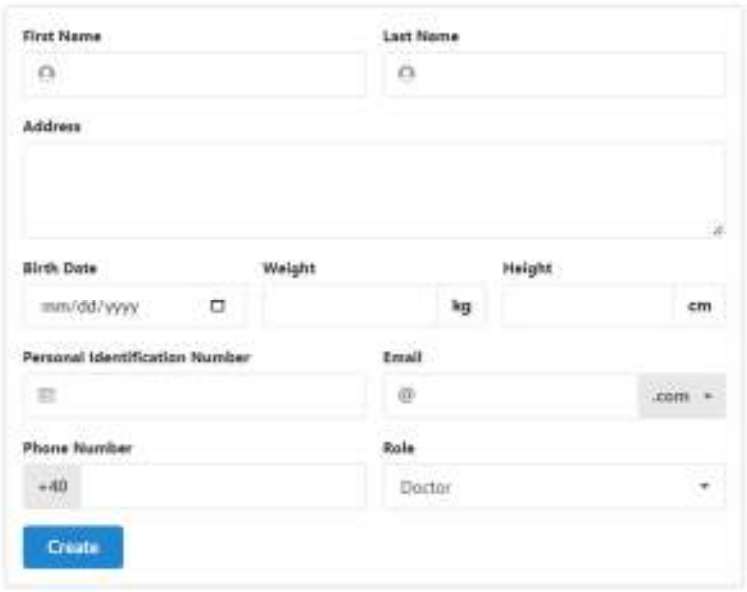

Fig. 4.20 Formularul de creare a unui utilizator

În funcție de rolul ales sunt executate și o serie de operații suplimentare, astfel:

- dacă utilizatorul curent este doctor, statusul privind posibilitatea de exercitarea a profesiei este verificat cu ajutorul Colegiul Medicilor din România, care pune la dispoziție un registru public cu datele medicilor din teritoriu;
- dacă utilizatorul curent este pacient, se va crea și un contract inteligent asociat care să reprezinte registrul medical al acestuia, prin intermediul căruia vor fi puse la dispoziție operațiunile de diagnosticare, cerere a datelor și vizionare a istoricului medical.

Modul de autorizare al operațiilor se realizează prin intermediul sesiunii gestionate de biblioteca Moralis, cât și de semnare a tranzacțiilor care adaugă date în blockchain cu ajutorul cheilor private. Astfel, orice operație menită să facă modificări asupră stării contractului inteligent, va determina apariția unei ferestre ce pune la îndemînă posibilitatea de confirmare sau refuzare a tranzacției. Întrucât nu este folosit un blockchain privat, ci rețeaua de test a Ethereum, operațiile au un timp mediu de realizare de 10-15 secunde. Pentru a sugera această așteptare utilizatorului, sunt folosite mecanisme vizuale de încărcare, cât și alerte cu mesaje intuitive atunci când o tranzacție s-a încheiat cu succes. Redirectarea este de asemenea folosită, pentru a genera atât reîncărcarea datelor noi adăugate, pentru mascarea timpilor de descărcare a actualizărilor, cât și pentru fluidizarea fluxului de lucru.

4.7.3. Colectarea datelor

Pentru colectarea datelor este necesară în primul rând prezența unui furnizor care să medieze interacțiunea server și rețeaua publică de blockchain. Logica din

spatele acestui furnizor poate fi regăsită în fișierul “web3.js”, în care fie preluăm furnizorul pus la dispoziție de Metamask dacă acesta este disponibil, sau utilizăm unul prestart de un serviciu public precum Infura care facilitează procesul de dezvoltare a aplicațiilor descentralizate. Trebuie avut în vedere că un astfel de serviciu nu include și capabilitățile unui portofel digital, astfel capacitatea de semnare a tranzacțiilor nu poate fi efectuată.

În al doilea rând, este nevoie de adresa la care a fost emis contractul inteligent. Librăria Web3 pune capabilități de a crea o instanță locală a acestui contract, de a apela metodele asociate și de a persista datele dorite pe rețeaua corespunzătoare, iar în plus fișierele “factory.js” și “registry.js” elimină redundanța acestei secvențe de cod.

```
componentDidMount = async () => {
    const { address } = this.props;
    const user = await factory.methods.users(address).call();
    const result = await factory.methods.retrieveRegistryRequests(address,user.privileges).call();

    const requests = await Promise.all(
        Array(parseInt(result[0])).fill().map(async (element, index) => {
            const id = await factory.methods.retrieveRequestIndex(index).call();
            const request = await factory.methods.retrieveRequest(id).call();
            const recordAddress = await factory.methods.deployedRegistries(request[2]).call();
            return {request, recordAddress};
        })
    )

    this.setState({ requests: requests, userPrivileges: user.privileges });

}
```

Fig. 4.21 Obținerea cererilor de acces

Maniera în care a fost abordată colectarea datelor, implică utilizarea operațiilor asincrone și a promisiunilor. Aceste concepte permit ca la momentul accesării unei pagini să poată fi instanțiat contractul dorit în funcție de parametrii de rutare, iar apoi generarea unui vector de promisiuni pentru fiecare parametru dorit. Prin sintaxa utilizată, starea componentei React va fi actualizată după finalizarea tuturor operațiilor și astfel vor fi încărcate toate rezultatele în același timp. În continuare nu sunt necesare operații de prelucrare asupra datelor, întrucât orice obiect rezulat din apelarea metodelor asupra contractului inteligent este în format JSON.

5. INTERACȚIUNEA UTILIZATOR-SISTEM

Interacțiunea utilizatorilor cu sistemul creat se va realiza doar prin intermediul interfeței Web, scopul acesteia fiind de a pune la dispoziție un număr de funcționalități care corespunde capabilităților principale oferite de componentele conexe. Pentru reprezentarea acestora, s-au constituit diagramele cazurilor de utilizare în funcție de rol, după cum urmează:

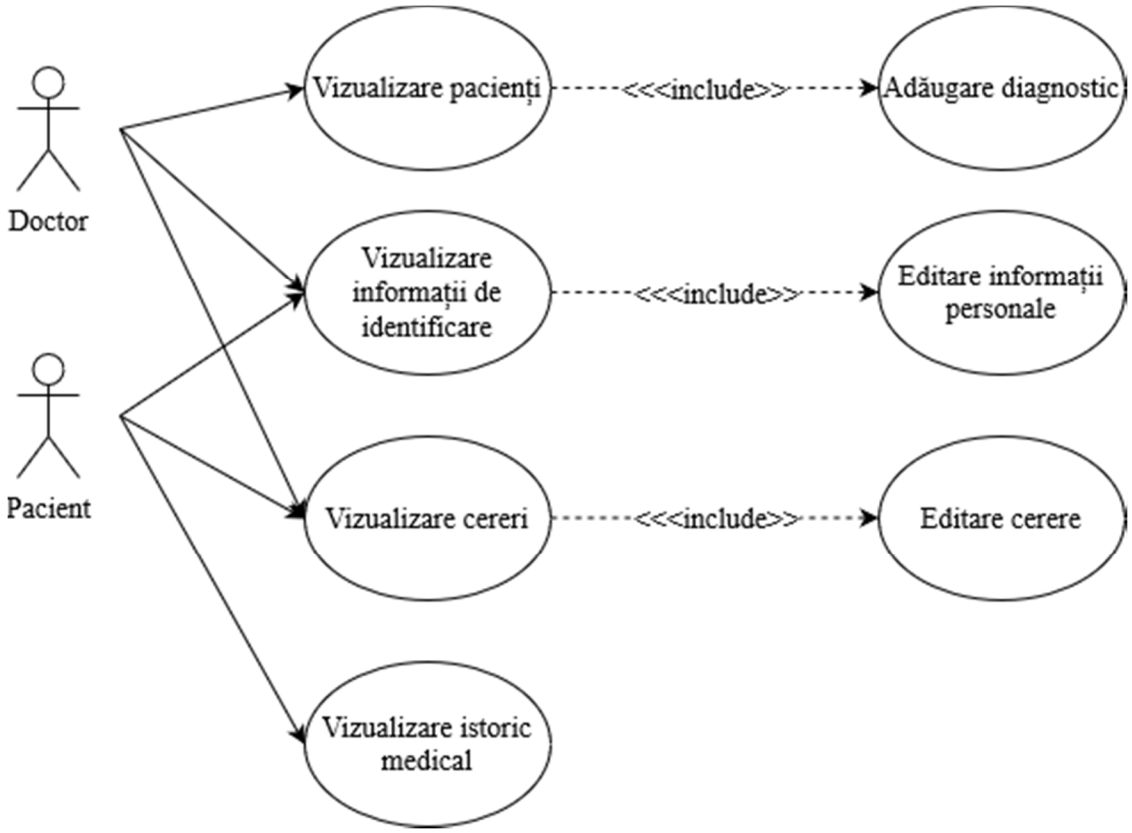

Fig. 5.1 Diagrama cazurilor de utilizare

Fluxul comunicării între microserviciile de stocare a datelor în cadrul blockchain-ului, cât și cel de consultare a modelului obținut prin procesul de învățare federată, este reprezentat în cadrul diagramei de secvență din figura ce urmează:

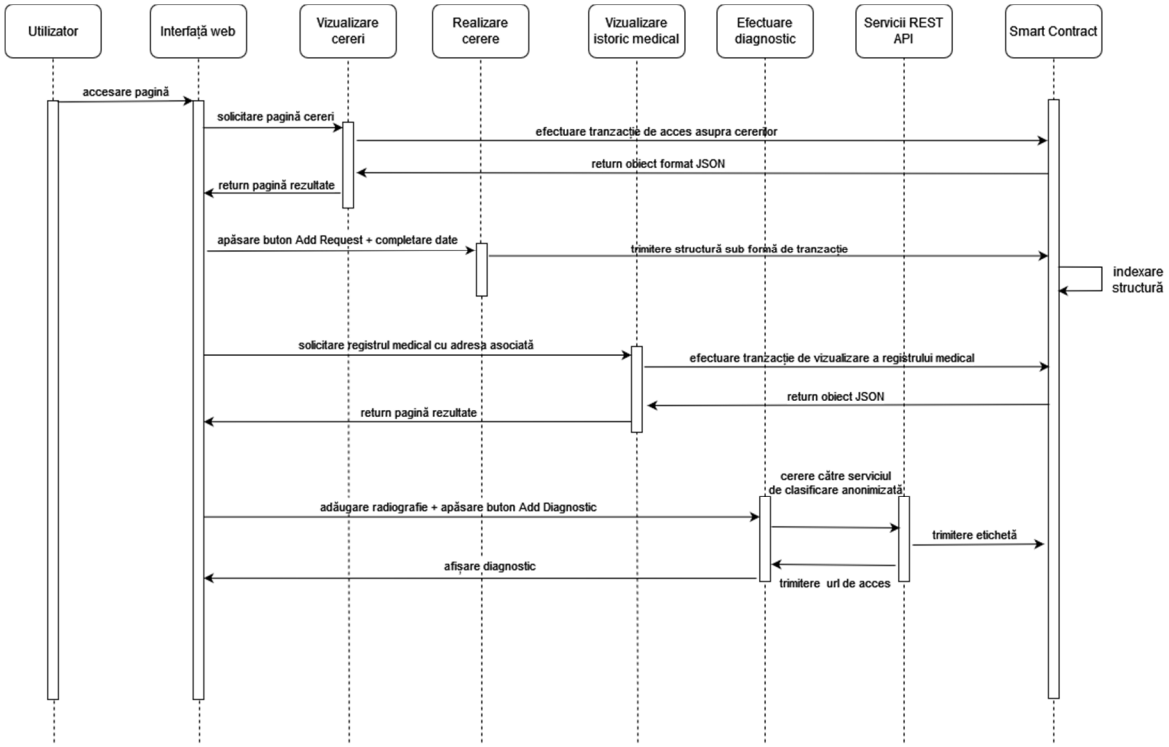

Fig. 5.2 Diagramă de secvență

Pagina inițiala a platformei Web conține reprezentarea detaliată a informațiilor asociate unui cont, cât și capacități de actualizare sau editare a datelor respective. Totodată, aceasta conține și rutele de acces către funcționalitățile de vizualizare a istoricului medical și de generare a unei cereri.

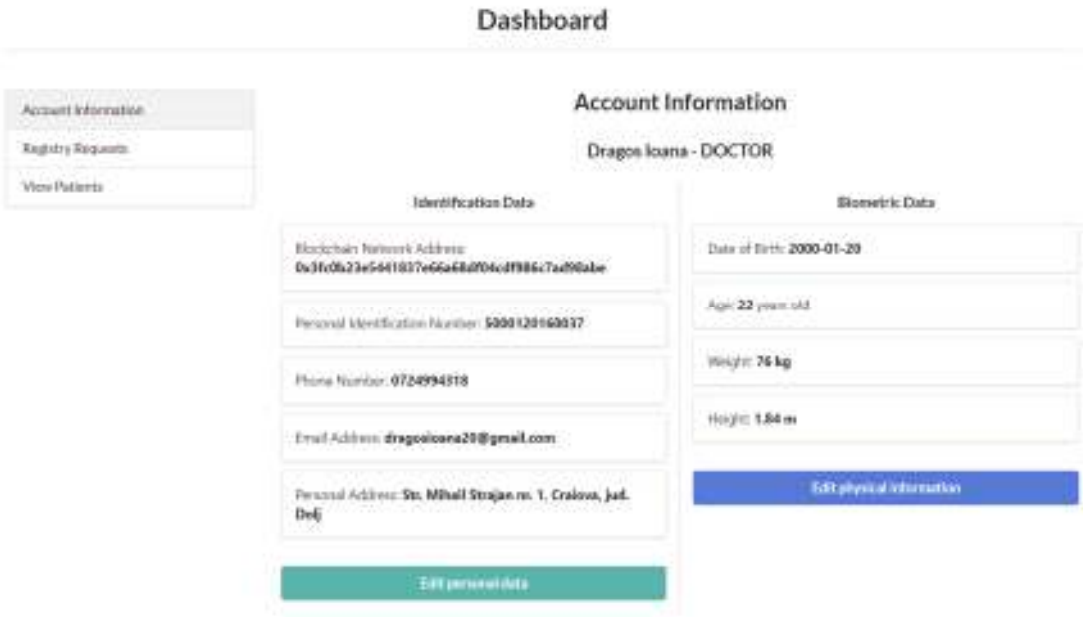

Fig. 5.3 Pagina de dashboard

În cadrul platformei Web, butoanele de navigare din partea stângă diferă ca și funcționalitate în funcție de rolul pe care îl are utilizatorul. Astfel aceste medoficări se manifestă fie prin vizualizarea sub formate sau de informații diferite, cât și pagini diferite. Doctorii au posiblitatea de generare a unei cereri de acces și prezintă un tabel cu fiecare cerere generată și statusul acesteia, în timp ce pacienții pot vedea toate cererile primite și decide modul de tratare al acestora.

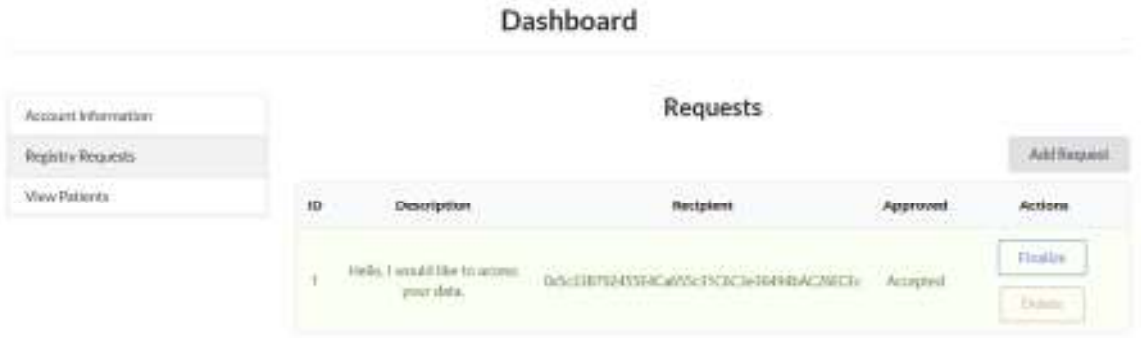

Fig. 5.4 Tabel cu cererile generate de un anumit doctor

Pentru fiecare cerere completată din tabel, apăsarea butonului de finalizare va conduce la pagina registrului medical al utilizatorului respective, unde sunt puse la dispoziție capacități de paginare a elementelor, cât și de adăugare a unui diagnostic, care se reduce la încărcarea unei radiografii trimisă către serviciul de anonimizare realizat prin antrenarea modelului și expunerea acestuia pe un server centralizat.

Fig. 5.5 Vizualizarea istoricului medical

De asemenea, doctorul are posbilitatea de vizualizare a unei liste, cu capabilități de paginare, care conține toți pacienții ce i-au conferit drepturi de acces.

Fig. 5.6 Vizualizarea pacienților

6. CONCLUZII

Soluția propusă pentru problematica abordată poate fi regăsită pe repository-ul public de git al autorului acestei lucrări. În cadrul acestuia, va fi detaliată atât arhitectura aplicației, cât și pașii ce pot fi urmați pentru recrearea mediului și utilizarea locală a implementării realizate. La momentul publicării și finalizării pașilor de inițializare a proiectului, va fi disponibilă o platformă web care să permită interacțiunea cu contractul inteligent emis pe rețeaua Rinkeby a blockchain-ului Ethereum, cât și utilizarea modelului preantrenat.

6.1. Probleme întâmpinate în implementare

Alegerea mecanismului de anonimizare a reprezentat prima problema îmtâmpinată, dată fiind natura datelor cu caracter medical. Deoarece, pe Internet există un număr foarte restrâns de seturi de date în format tabular și la acestea accesul trebuie autorizat după revizuirea unei cereri de către o comisie dedicată, iar colecția și etichetarea datelor se face de către personae cu cunoștințe de specialitate în domeniu, mecanismul de generare a datelor sintetice nu era unul fezabil.

Astfel o primă provocare a reprezentat-o alegerea unui mecanism care să respecte normele legale și reglementările în vigoare privind protecția datelor personale, cât și identificarea unui set de date care să poată fi utilizat. Soluția găsită s-a folosit de numeroasele seturi de date deindentificate de radiografii care pot fi folosite în antrenarea unui model de clasificare, la care se adaugă modulul de confidențialitate diferențială al cărui rol este de a putea oferi informații utile la nivel de populație, dar să nu permită identificarea persoanelor individuale. Argumentul alegerii acestei metode, este pe de-o parte formalismul matematic care stă la baza calculelor privind confidențialitatea care poate fi verificat, cât și demonstrate, iar pe de altă parte, numeroasele implementări contemporane care utilizează acest mecanism statistic la nivel macro.

Cea de-a doua problemă întâmpinată este cea a modului de creare a unei rețele de învățare federate pentru antrenarea sigură a unui model global agregat. Această parte este inclusă în pipeline-ul de anonimizare, întrucât datele colectate la nivel de spital sau clinică nu sunt partajate, ci doar modelele antrenate cu ajutorul acestora în format criptat. Pentru rezolvarea acestei probleme a fost identificat un framework ce permite conectarea prin intermediul unui server centralizat care necesită anumiți pași de configurare, a mai multor dispozitive și efectuarea de calcule la distanță.

6.2. Rezultate obținute

În urma implementării, aplicația înglobează trei module principale:

- un sistem de anonimizare, care realizează atât partea de antrenare locală și agregare a modelelor, pune la dispoziție un API pentru utilizarea clasificării și permite și managementul utilizatorilor participanți în rețea;
- platformă Web folosită pentru vizualizarea datelor persistate în blockchain și care pune la dispoziție și funcționalitățile unui registru electronic medical;
- o componentă de stocare, materializată sub forma unui contract inteligent cu mecanisme de control a accesului la metodele sale.

Un ultim rezultat îl reprezintă mecanismul de diagnosticare care îmbină atât funcționalitatea de introducere a unei imagistici medicale, utilizarea ei pentru generearea unei radiografii anonimizate, cât și stocarea pe blockchain a unui link de acces furnizat de un serviciu de cloud public asupra rezultatului.

6.3. Direcții viitoare

Îmbunătățirea care ar putea fi aduse proiectului este reprezentată de implementarea unui mecanism de tip “thread pool” la nivelul serverului centralizat care să se ocupe cu obținerea rezultatului din pipeline-ul de anonimizare. Pe lângă acest lucru, se poate folosi un serviciu de cloud public pentru mașinile pe care se va realiza antrenarea modelelor locale.

O funcționalitate suplimentară ar fi îmbinarea capabilităților de la nivelul interfeței Web pusă la dispoziție de framework-ul destinat învățării federate, pentru a se putea realiza și rolul de administrator cu capacități de management al utilizatorilor existenți.

7. BIBLIOGRAFIE

[1] Q. Government. [Interactiv]. Available: https://www.rti.qld.gov.au/information-privacy-act. [Accesat 10 06 2022].
[2] C. Europeană. [Interactiv]. Available: https://gdpr-info.eu/. [Accesat 11 03 2022].
[3] A. Government. [Interactiv]. Available: https://www.oaic.gov.au/privacy/the-privacy-act. [Accesat 21 06 2022].
[4] I. N. Y. X. G. H. S. G. ABID MEHMOOD, „Protection of Big Data Privacy," IEEE Access, 2016.
[5] L. Arbuckle și K. E. Emam, Building an Anonymization, O'Reilly, 2020.
[6] C. Europeană. [Interactiv]. Available: https://ec.europa.eu/research/participants/documents/downloadPublic?documentIds=080166e5b7698c70&appId=PPGMS. [Accesat 21 06 2022].
[7] T. L. S. V. Ninghui Li, „t-Closeness: Privacy Beyond k-Anonymity and l-Diversity," 2007.
[8] A. R. Cynthia Dwork, The Algorithmic Foundations, 2014.
[9] [Interactiv]. Available: https://differentialprivacy.org/reconstruction-theory/. [Accesat 12 06 2022].
[10] D. K. J. A. ,. J. G. ,. L. V. Ashwin Machanavajjhala, „Privacy: Theory meets Practice on the Map," 2008.
[11] S. Nakamoto, „Bitcoin: A Peer-to-Peer Electronic Cash System," 2008.
[12] W. Chen, „A Survey of Blockchain Applications in Different Domains," 2014.
[13] A. A. Fakhar ul Hassan, „Blockchain and the Future of the Internet: A Comprehensive Review," 2020.
[14] „Semantic Scholar," [Interactiv]. Available: https://www.semanticscholar.org/paper/An-Overview-of-Blockchain-Technology%3A-Architecture%2C-Zheng-Xie/ee177faa39b981d6dd21994ac33269f3298e3f68. [Accesat 23 06 2022].
[15] „Research Gate," [Interactiv]. Available: https://www.researchgate.net/publication/321322377/figure/fig2/AS:631670612955171@1527613418720/Blockchain-structure.png. [Accesat 25 06 2022].
[16] D. T. Wenbo Wang, „A Survey on Consensus Mechanisms and Mining Strategy Management in Blockchain Networks," 2019.
[17] IEEE, „An Overview of Smart Contract: Architecture, Applications, and Future Trends," [Interactiv]. Available: https://ieeexplore.ieee.org/document/8500488. [Accesat 13 06 2022].
[18] „NVIDIA Developer," [Interactiv]. Available: https://developer-blogs.nvidia.com/wp-content/uploads/2020/11/clara-train.png. [Accesat 15 06 2022].
[19] H. B. M. E. Moore, „Communication-Efficient Learning of Deep Networks from Decentralized Data," 2017.

[20] S. V. Brian Knott, „CRYPTEN: Secure Multi-Party Computation Meets Machine Learning,” 2021.
[21] „SUNFISH Platform,” [Interactiv]. Available: https://sunfish-platform-documentation.readthedocs.io/en/latest/_images/smc-computation.png. [Accesat 25 06 2022].
[22] Y. Lindell, „Secure Multiparty Computation (MPC),” 2020.
[23] M. F. P. Muhammad Ayaz. [Interactiv]. Available: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC8367140/. [Accesat 20 06 2022].
[24] T. Desfontain. [Interactiv]. Available: https://desfontain.es/privacy/real-world-differential-privacy.html. [Accesat 21 06 2022].
[25] „GeeksforGeeks,” [Interactiv]. Available: https://media.geeksforgeeks.org/wp-content/uploads/20200224050909/nodejs2.png. [Accesat 25 06 2022].
[26] „GeeksforGeeks,” [Interactiv]. Available: https://media.geeksforgeeks.org/wp-content/cdn-uploads/20220131100129/Group-3-1.jpg. [Accesat 26 06 2022].
[27] „Next.js,” [Interactiv]. Available: https://nextjs.org/static/images/learn/foundations/next-app.png. [Accesat 25 06 2022].
[28] „LogRocket Blog,” [Interactiv]. Available: https://blog.logrocket.com/wp-content/uploads/2019/06/ssr-explanation.png. [Accesat 30 06 2022].
[29] „Better Programming,” [Interactiv]. Available: https://miro.medium.com/max/1352/1*ronMtzhop4EL70l8lItDGA.png. [Accesat 28 06 2022].
[30] „Towards Data Science,” [Interactiv]. Available: https://miro.medium.com/max/1400/1*Bbgj7VNT-01KD3U2lWXDgw.png.
[31] „ResearchGate,” [Interactiv]. Available: https://www.researchgate.net/publication/336642248/figure/fig1/AS:839151377203201@1577080687133/Original-ResNet-18-Architecture.png. [Accesat 30 06 2022].
[32] G. M. C. W. Kermany D, „ Identifying Medical Diagnoses and Treatable Diseases by Image-Based Deep Learning,” 2018.
[33] [Interactiv]. Available: https://github.com/undacmic/electronic-health-registry.
[34] „https://sdcpractice.readthedocs.io/en/latest/anon_methods.html,” [Interactiv].
[35] J. Peat, E. Elliott, L. Baur and V. Keena, Scientific writing: easy when you know how, 2013.
[36] „Semantic Scholar,” [Interactiv]. Available: https://www.semanticscholar.org/paper/An-Overview-of-Blockchain-Technology%3A-Architecture%2C-Zheng-Xie/ee177faa39b981d6dd21994ac33269f3298e3f68. [Accesat 23 06 2022].

8. ANEXE

8.1. ANEXA A

```
contract ElectronicHealthRegistry {
    enum UserRole { ADMIN, DOCTOR, PATIENT}
    struct User {…
    }
    struct Request {…
    }
    modifier isDoctor(address user) {…
    }
    modifier isPatient(address user) {…
    }
    mapping(address => User) public users;
    address[] public userIndex;
    mapping(address => address) public deployedRegistries;
    mapping (string => Request) private requests;
    mapping (uint => string) private requestIndex;
    uint public requestCount;
    function isRequest(string calldata id) public view returns(bool) {…
    }
    function createUser(string memory _firstName,…
    }
    function getPatients(address doctor) public isDoctor(doctor) view returns(address[] memory){…
    }
    function createRegistry(address patient) public {…
    }
    function createRequest(string calldata description, string calldata id, address recipient)…
    }
    function retrieveRequestIndex(uint id) public view returns(string memory) {…
    }
    function retrieveRequest(string calldata id)…
    }
    function retrieveRegistryRequests(address patient, uint side)…
    }
    function approveRequest(uint index) public {…
    }
    function deleteRequest(string calldata id) public returns(uint) {…
    }
}
```

8.2. ANEXA B

```
contract HealthRegistry {
    struct HealthEntry {···
    }
    address public patient;
    mapping (address => bool) private doctors;
    mapping (string => HealthEntry) private entries;
    mapping (uint => string) private entryIndex;
    uint public entriesCount;
    modifier restricted() {···
    }
    modifier onlyPatient() {···
    }
    modifier present(string calldata uuid) {···
    }
    constructor(address person) {···
    }
    function retrieveEntry(string calldata uuid)···
    }
    function retrieveEntryIndex(uint index)···
    }
    function createEntry(string calldata uuid, uint data)···
    }
    function updateEntry(string calldata uuid, uint newData)···
    }
    function deleteEntry(string calldata uuid) public present(uuid) returns(uint) {···
    }
    function addDoctor(address doctor) public {···
    }
}
```