UNIVERSITATEA "ALEXANDRU IOAN CUZA" DIN IAȘI

FACULTATEA DE INFORMATICĂ

LUCRARE DE LICENȚĂ

Medchain - aplicație a tehnologiei blockchain în domeniul medical

propusă de

Răzvan Romănescu

Sesiunea: *februarie, 2019*

Coordonator științific

Prof. Colab. Florin Olariu

UNIVERSITATEA "ALEXANDRU IOAN CUZA" DIN IAȘI

FACULTATEA DE INFORMATICĂ

Medchain - aplicație a tehnologiei blockchain în domeniul medical

Răzvan Romănescu

Sesiunea: *februarie, 2019*

Coordonator științific

Prof. Colab. Florin Olariu

Avizat,

Îndrumător Lucrare de Licenţă

Titlul, Numele şi prenumele

_____

Data _____ Semnătura _____

DECLARAȚIE privind originalitatea conținutului lucrării de licență

Subsemnatul(a) …….

cu domiciliul în …….…….

născut(ă) la data de ……..…., identificat prin CNP ……..……..……..., absolvent(a) al(a) Universității „Alexandru Ioan Cuza" din Iași, Facultatea de ……. specializarea ……, promoția ……., declar pe propria răspundere, cunoscând consecințele falsului în declarații în sensul art. 326 din Noul Cod Penal și dispozițiile Legii Educației Naționale nr. 1/2011 art.143 al. 4 și5 referitoare la plagiat, că lucrarea de licență cu titlul:

_____

_____

_____

_____elaborată sub îndrumarea dl. / d-na _____, pe care urmează să o susțină în fața comisiei este originală, îmi aparține și îmi asum conținutul său în întregime.

De asemenea, declar că sunt de acord ca lucrarea mea de licență să fie verificată prin orice modalitate legală pentru confirmarea originalității, consimțind inclusiv la introducerea conținutului său într-o bază de date în acest scop.

Am luat la cunoștință despre faptul că este interzisă comercializarea de lucrări științifice în vederea facilitării falsificării de către cumpărător a calității de autor al unei lucrări de licență, de diploma sau de disertație și în acest sens, declar pe proprie răspundere că lucrarea de față nu a fost copiată ci reprezintă rodul cercetării pe care am întreprins-o.

Dată azi, …… Semnătură student ……

DECLARAȚIE DE CONSIMȚĂMÂNT

Prin prezenta declar că sunt de acord ca Lucrarea de licență cu titlul „*Medchain - aplicație a tehnologiei blockchain în domeniul medical*”, codul sursă al programelor și celelalte conținuturi (grafice, multimedia, date de test etc.) care însoțesc această lucrare să fie utilizate în cadrul Facultății de Informatică.

De asemenea, sunt de acord ca Facultatea de Informatică de la Universitatea „Alexandru Ioan Cuza” din Iași, să utilizeze, modifice, reproducă și să distribuie în scopuri necomerciale programele-calculator, format executabil și sursă, realizate de mine în cadrul prezentei lucrări de licență.

Iași,

Absolvent *Răzvan Romănescu*

_____

Cuprins

Introducere

Lucrarea de față își propune să evidențieze aspectele teoretice și practice, respectiv impactul și utilitatea pe care tehnologia *blockchain* o poate avea în viața de zi cu zi, avantajele pe care această tehnologie ni le oferă, limitările ei și ce soluții avem pentru a depăși aceste considerente, făcând această evidențiere prin implementarea unei aplicații ce se folosește de această tehnologie.

Latura practică a lucrării mele, aplicația, își dorește a fi un instrument ce are ca scop păstrarea integră a istoricului medical al pacienților de pretutindeni, cu un mod intuitiv și minimalist de interacțiune între entitățile implicate în acest proces. Această aplicație ar putea fi adoptată de orice țară dorește să treacă de la costurile și responsabilitatea dezvoltării unui sistem propriu de păstrare în siguranță și interacțiune eficientă cu istoricul medical al cetățenilor, la o soluție bazată pe tehnologia *blockchain ce* face acest lucru prin însuși mecanismele și arhitectura ei.

Una din problemele pe care aplicația mea își dorește să o rezolve este dificultatea, atât ca timp, cât și ca proces, accesării propriului istoric medical spre a fi mai bine informat si a lua decizii înțelepte în ceea ce privește propria sănătate și stil de viață. De asemenea, o altă problemă pe care aplicația și-o propune spre a fi rezolvată este păstrarea integră, nealterată și în siguranță a datelor medicale, întrucât ușurința cu care se pot pierde aceste date, cu o valoare semnificativă pentru fiecare individ, este îngrijorătoare.

Aceste probleme sunt recurente în majoritatea statelor lumii, întrucât nu există o soluție generică și aplicabilă pe întregul glob. Chiar și cu avântul globalizării, dacă ai nevoie de servicii medicale ce generează o formă de istoric medical (vizită periodică, operații chirurgicale, etc.) într-o altă țară, ești dependent și trebuie să te supui sistemului implementat în respectiva țară, istoricul rămânând în cadrul instituției medicale ale căror servicii le-ai solicitat.

Astfel, se produce fragmentarea istoricului medical al pacientului, istoric ce poate fi extrem de benefic în diverse situații ce necesită luarea de decizii în ceea ce privește sănătatea. Această fragmentare se poate realiza cu ușurință și într-o țară ce nu are un sistem medical bine pus la punct, nu dispune de o formă bine stabilită de centralizare a datelor medicale ale pacienților, ori nu are resurse să implementeze un astfel sistem.

Soluția la aceste probleme este o aplicație ce se folosește de avantajele pe care le oferă tehnologia *blockchain*, în detrimentul unei abordări clasice, centralizate.

Un "*blockchain*" este o serie de tranzacții într-o continuă creștere, dispuse în blocuri ce încapsulează aceste tranzacții, legate între ele folosindu-se principii și unelte criptografice în cadrul unei rețele P2P (peer-to-peer).

Printre proprietățile *blockchain*-ului se enumeră păstrarea permanentă a datelor (ce sunt dispuse sub formă de tranzacții), imposibilitatea alterării datelor odată scrise în cadrul registrului de tranzacții (astfel păstrându-se integritatea acestora), transparență (registrul de tranzacții este public, nu există tranzacții ascunse) și pseudo-anonimitate (autentificarea/autorizarea și interacțiunea în cadrul *blockchain*-ului se face prin adrese ce sunt constituite dintr-o cheie publică și una privată. Identitatea ta reală nu este legată de adresă în mod direct, dar prin natura transparentă a registrului de tranzacții se pot face inferențe legate de cine și cum utilizează *blockchain*-ul). Mai pe larg voi discuta despre tehnologia *blockchain* în capitolele ce urmează.

Putem observa din proprietățile unui *blockchain* modul cum s-ar putea plia în diverse arii, precum domeniul financiar, de business și chiar domeniul medical. Aceste proprietăți au stat la baza deciziei de a înfăptui un proiect folosind această tehnologie.

Seria de probleme expuse anterior legate de registrele de istoric medical au ca soluție o aplicație ce oferă unelte atât pacienților de a-și vizualiza și asigura integritatea datelor, cât și entităților medicale responsabile cu scrierea de date medicale sau a formelor de guvernământ (țărilor) de a-și face managementul entităților medicale ce își desfășoară activitățile în cadrul respectivei țări.

Motivele principale pentru care am ales această tematică au fost, pe de o parte, elementul de noutate și perspectivă nouă pe care tehnologia *blockchain* o poate aduce în viețile oamenilor și faptul că însuși natura tehnologiei se pliază excelent pe o serie de probleme cu care ne confruntăm zilnic. Pe de cealaltă parte, propria experiență cu sistemul medical de la noi din țară și cât de ușor este să-ți pierzi ani de zile de istoric medical scris pe hârtie, netranspus în format digital. Orice incident (incendiu, neatenția în operarea cu dosarele pacienților, etc.) poate duce la pierderea de date importante ce pot face diferența în situații ce necesită luarea de decizii pentru sănătatea unui pacient (un simplu lanț de înregistrări medicale de acum câțiva ani ar putea conduce către un diagnostic diferențial complet diferit în prezent, spre exemplu).

Un alt motiv pentru alegerea acestei teme a fost legată de accesibilitatea datelor. În aceeași notă cu ușurința prin care datele tale medicale pot fi pierdute, acestea sunt relativ greu de accesat printr-un mijloc eficient, mai ales daca acestea sunt în format fizic în cadrul unui dosar.

Astfel, am decis să simplific procesul plecând de la experiențele cu sistemul medical de la noi din țară, dar păstrând în minte ideea de globalizare și ușurință în accesarea datelor de oriunde ai fi. Idealul este ca de oriunde ai folosi servicii medicale să poți valida datele care îți vor fi scrise în istoricul medical și în același timp să fii tu deținătorul datelor tale, nu să fie păstrate într-un mod centralizat și dispersat în mii de locații. În același timp, din punct de vedere al țărilor și entităților medicale, am vrut ca aplicația sa fie simplă si extensibilă în funcție de situație (spre exemplu, daca o țară nu oferă servicii medicale celor care nu sunt cetățeni ai respectivei țări, atunci entitățile medicale nu vor aproba scrierea de istoric medical unui pacient ce provine dintr-o altă țară, etc.).

Dificultăți în dezvoltarea aplicației au apărut în urma faptului că tehnologia încă este tânără și în plină dezvoltare, existând o serie de limitări impuse de un astfel de registru distribuit de tip *blockchain* pentru care a trebuit să găsesc modalități să trec peste acestea, uneori reinventând roata sau repetând niște șabloane ce ar putea fi incluse în mod standard în limbajele suportate.

Spre exemplu, operații sau construcții simple și naturale din limbaje de programare consacrate nu se regăsesc în mod direct în limbajele suportate de tehnologia *blockchain* folosită, pe de o parte ca limitare din designul tehnologiei și implicit a limbajului, iar pe de cealaltă parte ca limitare datorată faptului că încă nu s-au implementat astfel de construcții relativ simple (e.g. returnarea unui tip de dată complex, precum un struct, în cadrul unei funcții).

Astfel, lucrarea de față este distribuită sub următoarea formă:

Primul capitol al acestei lucrări este cel al contribuțiilor, în care voi expune în mare cu ce elemente de noutate vine aplicația dezvoltată adiacent acestei lucrări.

Al doilea capitol se concentrează pe „Aspecte conceptuale” și pune în evidență elemente de natură teoretică, conceptele ce stau la baza tehnologiei *blockchain*, cum se descrie un *blockchain*, sumar cum se realizează consensul în cadrul acestei tehnologii, vom vedea cu ce se diferă prima generație de *blockchain* cu cea de a doua, ce limitări au ambele generații de tehnologii blockchain și ce posibile soluții (de altfel, aplicate în cadrul aplicației) avem în a trece peste aceste limitări.

Al treilea capitol este folosit spre evidențierea aspectelor arhitecturale și practice din cadrul proiectului dezvoltat. Elemente precum o descriere generală a aplicației și ce își propune să facă, arhitectura utilizată în cadrul dezvoltării acesteia, cum se pliază în comparație cu o

aplicație normală, clasică, dispunerea componentelor acesteia și descrierea lor cu o paralelă adusă către o arhitectură bazată pe microservicii și, sumar, câteva tehnologii și unelte noi ce au fost folosite în cadrul proiectului.

Al patrulea capitol este rezervat pentru „Concluziile lucrării”, unde prezint ce am învățat de pe urma dezvoltării acestei lucrări, respectiv a aplicației, direcțiile de viitor ale acesteia și cum ar putea fi îmbunătățită în viitor.

1. Contribuții

Proiectul de față, **Medchain**, își propune să fie o aplicație ce se folosește de infrastructura oferită de platforma *blockchain*, Ethereum, spre a oferi un mod simplu și intuitiv prin care oamenii din întreaga lume pot ține evidența istoricului medical, îl pot verifica și interoga oricând în așa fel încât să fie mai bine pregătiți în eventuala luare de decizii ce țin de sănătate, ori în situații de urgență unde istoricul medical este de o importanță considerabilă. În același timp, aplicația este o unealtă și pentru țările ce oferă servicii medicale prin prisma entităților medicale ce își desfășoară activitatea pe teritoriile acestora.

Ca idee de ansamblu, se urmărește interacțiunea dintre pacienți, istoricul medical al acestora și entitățile medicale calificate spre a scrie istoric medical. Toate acestea fiind sub umbrela beneficiilor oferite de tehnologia blockchain, precum păstrarea integră a datelor și lipsa de volatilitate a acestora (datele sunt permanente și mereu accesibile pe rețea). Deși datele sunt mereu accesibile, unul din obiectivele aplicației a fost depășirea unei limitări a acestei tehnologii, faptul că datele sunt publice și ar putea fi accesate de oricine, prin mecanisme de autorizare și criptarea datelor în cadrul *blockchain*-ului, păstrând, astfel, intimitatea pacienților când vine vorba de date senzitive și detaliate.

Aplicația se diferențiază de tot ceea ce este pe piață la momentul actual prin faptul că este gândită drept o soluție generică ce poate fi folosită la scară globală, prin faptul că datele nu sunt păstrate într-un loc centralizat, nefiind, astfel, susceptibile spre pierdere, ci folosindu-se tehnologia blockchain ne folosim de o rețea distribuită, peer-to-peer, necontrolată de o autoritate centrală și mecanisme precum *smart contracts* de automatizare a interacțiunilor în cadrul rețelei. De regulă, soluțiile de acest gen tind să nu aibă întreaga interacțiune pacient – entitate medicală – scrierea istoricului pusă la punct, ci oferă în principal acces spre vizualizarea istoricului și este limitată exclusiv la istoricul medical generat pe teritoriul țării de proveniență.

Aplicația nu se limitează la o anumită țară, ci își propune să fie disponibilă oricărei țări interesate de un astfel de sistem sigur și intuitiv de management al istoricului medical. Tot ceea ce este necesar e ca respectiva țară să fie înregistrată în sistem de către autoritatea principală a platformei (autoritate ce poate fi schimbată), ca ulterior să-și poată adăuga entitățile medicale ce lucrează pe teritoriul său.

2. Aspecte conceptuale

În cadrul acestui capitol voi face o trece prin ceea ce este tehnologia blockchain, câteva detalii legate de cum funcționează aceasta, o serie de beneficii aduse de tehnologie, dar și limitările pe care le are, urmând ca apoi să descriu ce soluții am putea folosi pentru a le depăși.

2.1. Blockchain – privire de ansamblu

Pentru a înțelege de ce a apărut tehnologia blockchain și ce încearcă să rezolve, trebuie să urmărim contextul istoric care a determinat accelerarea dezvoltării acesteia.

O serie de evenimente neplăcute s-au înfăptui în jurul anului 2008, pe de o parte în spectrul american, pe de cealaltă parte la nivel global din punct de vedere financiar. De la criza economică mondială până la breșe de securitate în sistemele informatice bancare ce au expus milioane de carduri de credit și date private ale cetățenilor lumii, aceste tipuri de evenimente au alimentat sentimentul de repulsie față de sistemul bancar clasic. Întotdeauna a existat o repulsie față de nevoia de a avea o formă de autoritate centrală ce are ca scop validarea sau invalidarea tranzacțiilor pe care noi, cetățenii normali, le facem. Iar într-o lume din ce în ce mai conectată, nevoia acestei autorități centrale s-a accentuat și mai mult pentru a face conceptul de sistem financiar digital să funcționeze.

Astfel, la începutul anului 2009 un nou tip de infrastructură a fost făcut public, în cadrul căreia primele 50 de monede virtuale au fost înregistrate în cadrul unui registru de tranzacții aproape imposibil, din punct de vedere al fezabilității, de a fi fraudat (spre exemplu, de a adăuga tranzacții invalide sau ilicite), registru replicat în cadrul unei rețetele peer-to-peer de computere conectate la Internet. Aceste 50 de monede virtuale reprezintă primul bloc de tranzacții din cadrul blockchain-ului Bitcoin, denumit bloc de geneză, urmând ca toate tranzacțiile ulterioare să fie legate de acest prim bloc sub forma unui lanț, existând o ordine bine stabilită a blocurilor și imposibil de schimbat odată ce au fost verificate în cadrul rețelei. De aici și numele de „blockchain”.

2.2. Prima materializare a tehnologiei blockchain – Bitcoin

Problema inițială pe care blockchain-ul Bitcoin încerca să o rezolve era problema de „double-spending” în cadrul unui sistem informatic. Concret, orice resursă din mediul digital poate fi

copiată cu ușurință de către oricine altcineva are acces la respectiva resursă. Acest lucru este în regulă în contextul în care lucrăm cu fișiere, text și elemente ce au ca scop de a fi împărtășite către alții, dar nu se mai aplică atunci când vorbim de bunuri digitale precum fișiere muzicale licențiate unei persoane, drepturile sau actele de proprietate ale unei persoane pentru un anumit bun fizic și nu în ultimul rând, bani. Aceste ultime elemente au un proprietar, dar nu putem asigura certitudinea că acele bunuri au cu adevărat acel proprietar din moment ce acestea pot fi replicate cu ușurință.

Problema „double-spending" se rezumă la ideea că într-un astfel de sistem în care bunurile digitale pot fi replicate cu ușurință, există riscul ca o aceeași monedă virtuală să poată fi folosită de mai multe ori la efectuarea tranzacțiilor.

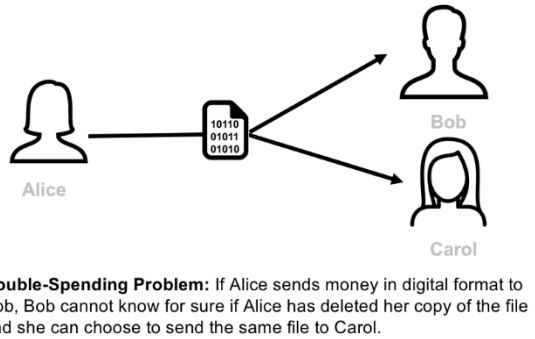

Figura 1 – Problema „double-spending"

Sursa: http://adilmoujahid.com/posts/2018/03/intro-blockchain-bitcoin-python/ (1)

Mecanismele prin care blockchain-ul Bitcoin rezolvă această problemă le voi descrie în capitolul legat de componentele de bază ale acestei tehnologii. La un nivel de ansamblu, blockchain-ul implementat inițial în cadrul Bitcoin taie nevoia de a avea o autoritate centrală, guvernatoare care să supervizeze toate tranzacțiile ce sunt înfăptuite pe rețea, în același timp oferind siguranța că monedele virtuale aparțin proprietarilor autentici și că nu există un mod fezabil de a frauda tranzacții sau de a schimba proprietarii acestor monede.

După cum exemplificam în subcapitolul anterior, tehnologia blockchain a luat naștere odată cu moneda Bitcoin, fiind primul exemplu practic al acestei tehnologii. Când vine vorba de blockchain-ul Bitcoin, acesta a fost gândit din start spre a furniza mecanismul necesar prin care banii virtuali pot circula liber și pot fi folosiți cu încredere în tranzacții digitale. În prezent, termenul de blockchain nu se rezumă strict la monede virtuale drept bunuri ce sunt urmărite în cadrul registrului, ci la orice fel de bunuri digitale (drepturi de autor, acte ce atestă cine este

proprietarul unui anumit imobil, etc.). Orice element din viața reală ce poate fi transpus în format digital poate să fie urmărit în cadrul unei rețele blockchain.

Ce face obiectul de studiu blockchain un lucru interesant este faptul că pe lângă ce rezolvă, acesta definește un protocol de interacțiune a entităților în cadrul rețelei și permite nodurilor acesteia să guverneze întreaga operațiune și să se guverneze singure, fiind limitate doar de regulile impuse de protocol. Astfel, se taie nevoia unui server central ce ține evidența a cine și ce face sau a vreo unei forme de autoritate, a unui om în mijloc, pentru a înfăptui orice tranzacție, în același timp păstrând nevoia de viteză și accesibilitate pe scară largă la această tehnologie.

2.3. Componente de bază a tehnologiei blockchain

Indiferent de tehnologia blockchain-ul la care ne referim (fie Bitcoin, Ethereum, ori cu totul și cu totul o altă adaptare a acesteia), aceasta are un set de componente de bază, regăsite în toate implementările tehnologiilor de acest tip. În cadrul acestui subcapitol voi face trecerea prin principalele componente ce asigură buna funcționare a unei rețetele blockchain.

2.3.1. Semnături digitale

Pentru a asigura faptul că o anumită tranzacție este făcută de o persoană și nu de oricine altcineva, avem nevoie de un mecanism simplu și rapid prin care să verificăm acest lucru. Tehnologia blockchain se folosește în mod extensiv de criptografia bazată pe cheie publică, model în care fiecare participant din cadrul rețelei se identifică printr-un set de chei, una publică, fiind identitatea către exterior, și una privată, care, după cum subliniază și numele, este o cheie personală ce rămâne cunoscută doar de către respectiva persoană.

Toate tranzacțiile au atașate semnătura digitală a emițătorului, generată prin luarea mesajului (conținutul tranzacției) și aplicarea unei funcții de hash cu cheia sa publică, generând, astfel, un șir de biți unic. Ulterior, în procesul de validare a unei tranzacții, este luat mesajul unde i se aplică o funcție de verificare a hash-ului, primind ca input cheia publică a respectivului individ și având ca output un răspuns afirmativ sau negativ, dacă semnătura digitală este produsă de cheia privată asociată cu cheia publică folosită în cadrul verificării.

O funcție hash nu este altceva decât o funcție de criptare într-o singură direcție. Mai concret, având ca date de intrare un mesaj, trecând acest mesaj printr-o funcție hash vom obține un șir

de caractere unic și de dimensiuni fixe. O proprietate a unei funcții hash este faptul că având output-ul acesteia, nu putem deduce computațional, în niciun mod, mesajul inițial. În schimb, dându-i acestei funcții același mesaj drept date de intrare, ne așteptăm să primim întotdeauna același hash (date de ieșire), subliniind, astfel, natura lor deterministă. De asemenea, două seturi de date de intrare diferite au o probabilitate extrem de mică de a da aceleași date de ieșire. În situația în care două astfel de seturi de date diferite generează același hash, numim acest incident drept o „coliziune”.

2.3.2. Arbori Merkle

O structură de date ce stă la baza tehnologiei blockchain, ce sunt definiți drept arbori de hash-uri. Fiecare nod copil al arborelui deține hash-ului unui bloc de date (putem să ne gândim la un fișier text, un fișier muzical, ori chiar un bloc de tranzacții în cadrului blockchain), în timp ce toate nodurile care nu sunt frunze sunt hash-uri ale concatenării hash-urilor nodurilor lor copil. În final, avem rădăcina arborelui ce reprezintă un hash al întregului arbore.

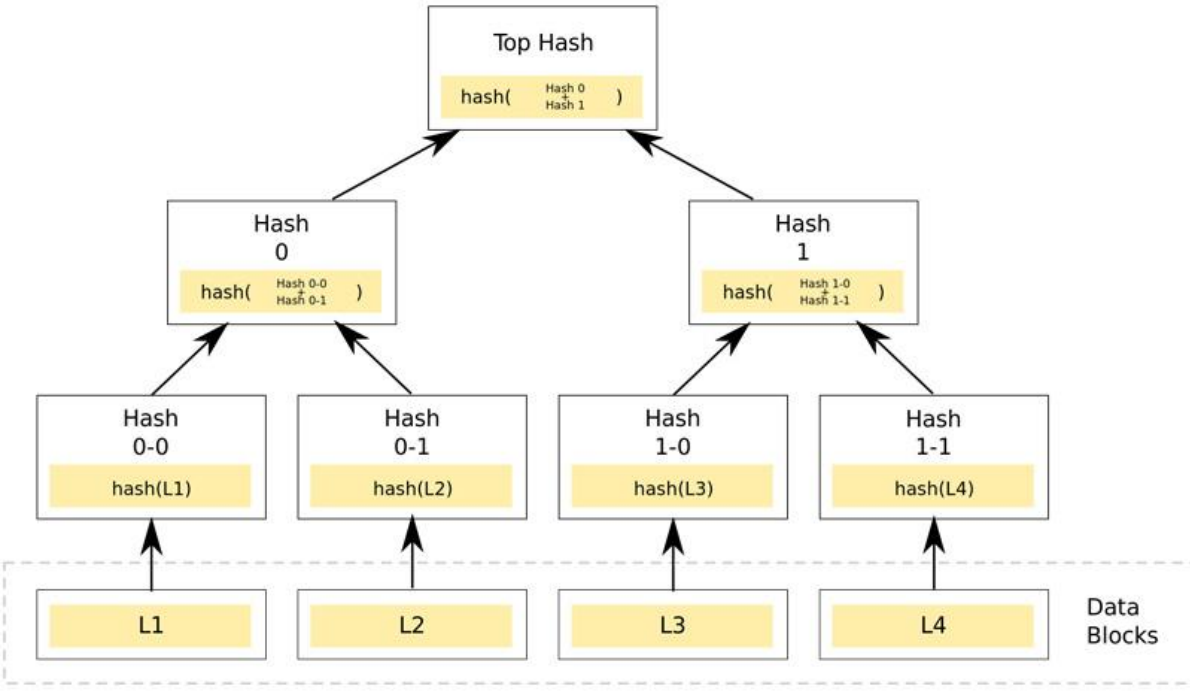

Figura 2 – Arbore Merkle binar

Sursa: https://en.wikipedia.org/wiki/Merkle_tree (2)

Arborii de tip Merkle sunt, ca privire de ansamblu, o reprezentare eficientă a unui set de date ce conferă niște proprietăți precum verificarea eficientă și securizată a modificărilor din cadrul setului de date, iar în contextul blockchain putem spune că permite scalarea acestui sistem, indiferent de creșterea dimensiunii setului de date (care, în cazul unui blockchain, poate să crească considerabil de mult). Dar în același timp, îi permite sistemului să-și păstreze arhitectura bazată pe mecanismul de hashing ce conferă integritatea datelor și un mod simplu de verificare a integrității acestora.

O proprietate interesantă a acestor arbori este faptul că dacă încerci să modifici o frunză, vei schimba toate hash-urile de deasupra sa și implicit nodul rădăcină. Arborii Merkle conferă un mod ieftin, din punct de vedere computațional, de verificare dacă o porțiune mică de date se află într-un set mare de date. În cadrul unei rețele distribuite, acest lucru conferă încredere între membrii respectivei rețele atunci când trebuie să mențină o bază de date actualizată și cu date integre.

În concluzie, în cadrul unui blockchain, blocurile de tranzacții se leagă între ele într-o anumită ordine, ordine conferită și ușor verificabilă de proprietățile unui arbore de tip Merkle, mecanism ce evită, astfel, problema „double-spending”.

2.3.3. Mecanism de consens

Ajungem, astfel, în cadrul acestei structuri de date menționate anterior, să avem nevoie de un mecanism prin care să decidem care este această rădăcină a arborelui în care să avem încredere ca fiind cea corectă. Aici intervine un mecanism ce oferă consens între membrii rețelei, mecanism ce decide cine urmează să adauge un nou bloc, un nou set de tranzacții în cadrul acestei baze de date într-o continuă creștere.

Atât blockchain-ul Bitcoin, cât și cel Ethereum ce urmează să fie prezentat ulterior, se folosesc de un algoritm ce conferă consens pe baza unei „dovezi a muncii” (proof of work). Acest algoritm se bazează pe ideea că un nod din cadrul rețelei trebuie să demonstreze faptul că are ca scop validarea unui astfel de bloc de tranzacții prin rezolvarea unei probleme computaționale dificile. Pentru a motiva nodurile din rețea să participe la un astfel de proces și consum de resurse cu scopul validării tranzacțiilor, acestora le este oferită o motivație financiară prin faptul că, daca reușesc să rezolve problema respectivă primii, aceștia vor avea dreptul să scrie noul bloc în istoricul blockchain-ului și implicit să obțină un anumit număr de monede virtuale

a respectivului blockchain (bitcoin, in cazul Bitcoin, ether în cazul Ethereum, etc.). Acest mecanism este o modalitate de evitare a asignării aleatorii a unui nod din rețea de a valida un anumit bloc de tranzacții.

Astfel, nevoia unui mecanism aleatoriu de selecție ce poate să nu fie atât de „aleatoriu" e asigurată de faptul că nu se știe ce nod va rezolva primul respectivul puzzle computațional, nicio autoritate centrală nu este necesară. De asemenea, blockchain-urile, prin natura lor, sunt descentralizate și conferă rezistență la a avea anumite noduri sau grupuri de noduri cu unicul scop de a fii singurii care să scrie blocuri în istoric, implicit dictând ordinea tranzacțiilor sau chiar natura lor. Ar fi necesară puterea computațională a 51% din întreaga capacitate a nodurilor de pe rețea, lucru extrem de infezabil.

2.4. Generația a doua blockchain – Ethereum

Ethereum face parte din a doua generație de blockchain și a apărut către publicul larg în anul 2013, plecând de la ideea că tehnologia blockchain poate să facă mai mult decât să țină evidența tranzacțiilor de valută virtuală.

Încă de la apariția Bitcoin au început să apară diverse abordări ale tehnologiei blockchain orientate spre aplicații dincolo de domeniul financiar (spre exemplu, un blockchain ce se axa pe furnizarea de servicii DNS, strângere de fonduri de tipul „crowd-funding", etc.). Limitarea acestor soluții blockchain era faptul că se concentrau pe oferirea a unui singur serviciu, un singur scop. Ulterior au început să apară blockchain-uri ce ofereau mai multe lucruri deodată. Evidența banilor, strângere de fonduri, etc. Fiecare categorie pe care o acopereau reprezenta un modul cu propriul cod și propria logică din cadrul blockchain-ului. Problema survine atunci când apare o nouă categorie ce blockchain-ul respectiv ar trebui sa o acopere, un nou tip de tranzacție pe care trebuie să o suporte, situație în care ar trebui creată o implementare nouă pentru respectiva categorie în cadrul protocolului, lucru nu foarte pragmatic.

Schimbarea, în cadrul Ethereum, e dată de privirea blockchain-ului drept o mașină, un computer în cadrul unei rețele și mai puțin drept o unealtă cu un singur scop, ori o multi unealtă ce necesită o nouă iterație pentru a include o nouă unealtă în setul ei standard.

Ethereum este un blockchain ce, pe lângă bagajul tehnologic pe care Bitcoin îl aduce, adaugă o serie de elemente noi ce îl diferențiază de cel din urmă. Un astfel de element nou este un

limbaj de programare specific pentru Ethereum ce rulează în cadrul unei mașini virtuale (Ethereum Virtual Machine).

2.5. Smart Contracts

Smart contracts sunt programe ce pot fi scrise spre a fi utilizate în cadrul blockchain-ului Ethereum. Aceste programe pot fi scrise în orice limbaj suportat de Ethereum (cel de bază se numește Solidity) și urmează o paradigmă orientată pe contracte.

Pentru a înțelege ce sunt contractele, avem nevoie de o delimitare a conturilor ce pot apărea pe Ethereum. Pe de o parte avem conturile utilizatorilor normali, ce se folosesc de perechile de chei publice și private și pot identifica un individ în cadrul rețelei, fiind controlate, de regulă, de factorul uman, iar pe de cealaltă parte avem contractele ce sunt controlate de regulile scrise în codul pe care acestea le dețin.

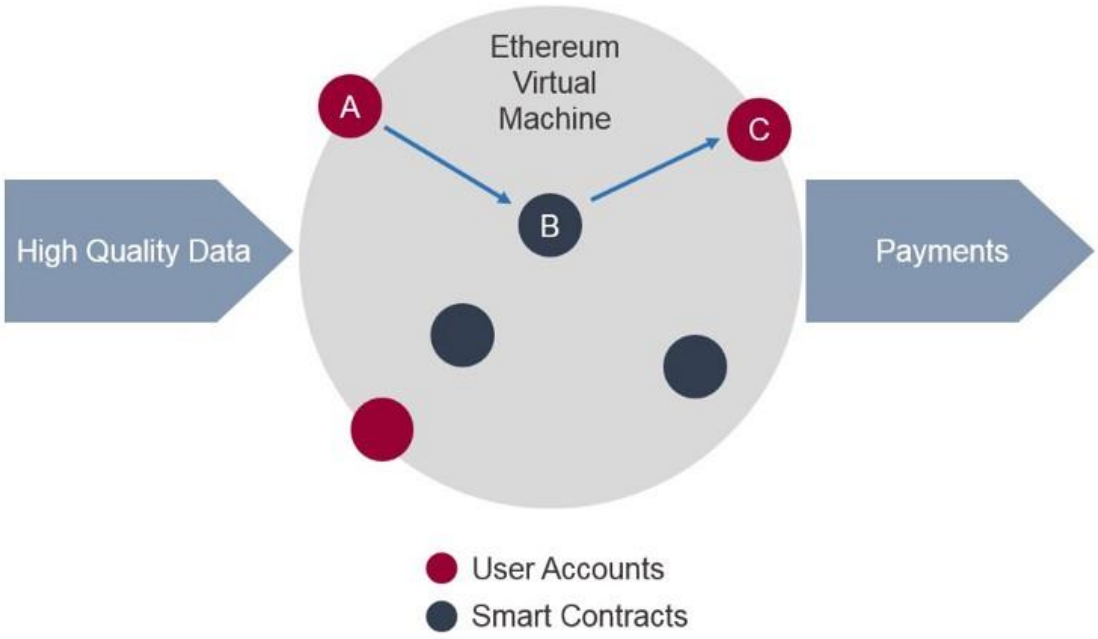

Figura 3 - Imagine de ansamblu asupra tipurilor de conturi de pe Ethereum

Sursa: https://medium.com/@fhansmann/smart-contracts-and-the-real-world-a-complicated-relationship-c00dd766731a (3)

În cadrul Ethereum păstrăm capabilitatea de a efectua tranzacții între conturile utilizatorilor, de a trimite monede virtuale dintr-un cont în altul (spre exemplu, de la contul A la contul C în mod direct), dar elementul de noutate este adus de faptul că utilizatorii pot interacționa cu

contractele de pe blockchain, iar interacțiunea este dictată de codul ce este executat în cadrul contractului cu pricina. Astfel, utilizatorul A poate interacționa cu contractul B, iar logica lui B poate să decidă, daca a fost programat în acest scop, să trimită o anumită porțiune din suma trimisă către el într-o altă parte, sau să schimbe anumite drepturi, etc. Practic orice poate fi modelat sub formă de cod de nivel înalt.

Contractele sunt o modalitate prin care poți programa regulile unei aplicații cu ajutorul codului ce poate fi scris într-un limbaj orientat pe contracte, ca ulterior să poți încărca o astfel de aplicație cu ajutorul unei tranzacții pe blockchain-ul Ethereum. Acestea pot fi considerate drept niște automate ce trăiesc în cadrul Ethereum, au o stare curentă a datelor, o zonă de stocare persistentă unde păstrează aceste date pe termen lung și un istoric, practic toate tranzacțiile care au interacționat și au avut legătură cu respectivul contract.

```
pragma solidity ^0.5.0;

contract Example {
    mapping(address => uint) private _balances;

    function deposit() public payable {
        _balances[msg.sender] += msg.value;
    }

    function withdrawable(uint amount) public view returns(bool) {
        if (amount < _balances[msg.sender]) {
            return true;
        }

        return false;
    }
}
```

Figura 4 - Exemplu de contract scris în Solidity

În **Error! Reference source not found.** putem observa un exemplu de contract ce se poate regăsi pe Ethereum. Acest contract are o funcție simplă și anume orice cont de utilizator care interacționează cu metoda „deposit” din acest contract, printr-o tranzacție, poate să depoziteze o anumită sumă de ether, iar cu ajutorul metodei „withdrawable” pot să verifice daca pot extrage o anumită sumă de ether. Putem observa construcții de limbaje de nivel înalt ce ne permit să modelăm logica unei aplicații destul de rapid și într-un mod familiar, dar aceste limbaje suportă și scrierea de cod de nivel jos, cod de asamblare pentru operații ce necesită performanță în cadrul contractelor.

Contractele pot fi considerate drept niște microservicii ce trăiesc pe blockchain-ul Ethereum. Acestea nu pot influența execuția altor contracte, ori nu pot efectua tranzacții în numele altor contracte, de aici și paralela cu microserviciile. Acestea sunt sisteme autonome ce înglobează o logică specifică ce pot fi executate de către alți utilizatori din cadrul blockchain-ului, având propria stare curentă, propria zonă de stocare și propriul istoric. Contractele așteaptă tranzacții ce interacționează cu ele, validează date, execută logica și formulează un răspuns în urma tranzacției. Cu toate acestea, un contract poate să aibă ca dependență instanța unui alt contract, astfel că logica unui astfel de contract depinde de logica respectivei dependințe. Dar chiar și în această situație, orice apel de funcție din contractul dependent către contractul părinte nu permite generarea de tranzacții în numele celui din urmă contract, ci pur si simplu deleagă o anumită porțiune de logică spre un alt contract.

Putem observa, astfel, că o simplă tranzacție a unui utilizator obișnuit către un contract poate să genereze o serie de alte tranzacții către o multitudine de alte contracte, acest lucru permițând desfășurarea de comportamente complexe în cadrul Ethereum.

2.6. Combustibil în Ethereum

Una din problemele pe care le ridică posibilitatea de a rula programe pe platforma Ethereum este „the halting problem”. Pe scurt, în timpul execuției unui program nu putem avea certitudinea că acel program își va finaliza execuția în timp util, ori va rula la infinit. Situația asta este neplăcută în contextul Ethereum, întrucât managementul eficient de resurse și putere de calcul este o necesitate.

Modul prin care Ethereum rezolvă această problemă este prin a introduce conceptul de combustibil („gas”) al unei tranzacții. Fiecare tranzacție are atașată o limită de combustibil pe care o poate suporta, acest combustibil fiind o cantitate ce se poate translata în moneda ether din cadrul Ethereum. Putem considera acest combustibil drept taxa rulării unui program. În cadrul unui contract, în momentul când primește o tranzacție ce apelează cod, mașina virtuală Ethereum va calcula fiecare pas computațional din cadrul respectivei porțiuni de cod apelat.

Fiecare operație în mașina virtuală Ethereum are un cost echivalent în combustibil. Spre exemplu, o operație poate fi scrierea persistentă a unor date într-o variabilă, parcurgerea și citirea unui vector, etc. Astfel, nu putem avea situații de genul unei bucle infinite în urma unei tranzacții, întrucât combustibilul atașat acesteia se va epuiza, tranzacția va eșua, orice ether

trimis în respectiva tranzacție se va restitui, dar costurile aferente execuției (combustibilul) nu se vor restitui.

Astfel, se asigură faptul că nodurile vor putea procesa în timp util și într-un mod determinist tranzacțiile pe care le primesc. De asemenea, acest mecanism are implicații la nivel de abordare legată de tranzacții. Tranzacții ce pornesc un proces computațional mai greu vor necesita taxe per tranzacție mai mari.

2.7. Avantajele tehnologiei blockchain

Tehnologia blockchain deschide porțile către o lume cu noi posibilități și conferă o serie de avantaje de care putem profita în dezvoltarea aplicațiilor de mâine.

Principalul avantaj al acestei tehnologii este transparența și faptul că efortul de dezvoltare este cu totul și cu totul open-source. Implicit, oricine s-ar putea folosi de implementările actuale pentru a crea ceva propriu, orientat spre nevoile anumitor companii, grupuri, ori indivizi. De asemenea, transparența acțiunilor și a istoricului generat în cadrul blockchain-urilor este augmentată de faptul că datele își păstrează integritatea și nu pot fi modificate în mod ilicit de către o persoană sau un grup. Acest lucru permite dezvoltarea de aplicații ce se concentrează pe oferirea de servicii dependente de date și siguranța lor.

Un alt avantaj al tehnologiei blockchain este securitatea prin design a acesteia. Față de scenariul clasic în care datele pe care le ai sunt ținute de către o formă de autoritate centrală predispusă la atacuri cibernetice, acestea sunt distribuite într-o rețea peer-to-peer, iar cu mecanisme de consens în spate fac blockchain-urile un loc sigur pentru aplicații ce au nevoie să protejeze datele utilizatorilor.

Poate cel mai interesant avantaj este faptul că tehnologia blockchain reduce pe cât de mult posibil factorul uman din cadrul tranzacțiilor. În timp ce în mod clasic o serie de operațiuni mânuite de un om pot să fie eronate și să producă daune, în situația blockchain-urilor lipsa de erori este dată de cât de bine este scris codul care automatizează procesele. Bineînțeles, și acestea din urmă pot suferi erori din cauza factorului uman, dar când vine vorba de cod există mecanisme de testare și perfecționare a acestuia, în timp ce natura umană rămâne impredictibilă.

De asemenea, când vine vorba de blockchain, toate bazele de date la care ne putem gândi și în cadrul cărora datele noastre există se pot comasa într-un singur registru ce deține toată informația într-un singur loc. Astfel, se reduce nevoia de a împrăștia datele în mai multe locuri, ca apoi să creăm mecanisme de comunicare între sisteme diferite, dar care dețin date comune despre noi. Totodată este mult mai ușor și rapid să ai un singur loc în care datele tale există.

Nu în ultimul rând, tehnologia blockchain are capabilitatea de a reduce costurile unei organizații, a dezvoltării și menținerii de software, pe de o parte pentru că se reduce nevoia de a avea nenumărați intermediari între organizație și utilizatori, iar pe de cealaltă parte, chiar dacă și în cadrul blockchain-urilor avem conceptul de taxe, de regulă acestea sunt mult mai accesibile decât costurile generate de dezvoltarea unui soft și menținerea propriei infrastructuri peste care aplicația să funcționeze.

2.8. Limitări ale blockchain-ului

Una din limitările pe care tehnologia blockchain le are, de altfel limitare de care m-am lovit și eu în cadrul dezvoltării aplicației **Medchain**, este faptul că stocarea persistentă de fișiere de dimensiuni mari nu este fezabilă într-un astfel de sistem, de altfel nu este nici recomandat să folosești Ethereum în acest sens. Acest lucru se datorează, pe de o parte, limitei de dimensiune a unei tranzacții ce este dictată de cantitatea de combustibil ce poate fi consumată într-o tranzacție, întrucât, pentru a trimite un fișier spre stocare pe blockchain este nevoie de o tranzacție. Iar cu cât sunt mai mari datele de intrare ale unei tranzacții, cu atât o să consume mai mult combustibil, implicit taxe mai mari ale tranzacției.

O altă limitare a acestei tehnologii este faptul că datele de intrare ale unei tranzacții sunt publice și accesibile de către oricine are interesul de a observa activitatea pe rețea. Mai concret, chiar daca am defini un contract ce ține în siguranță datele pe care le primește și nimeni neautorizat nu ar putea să le acceseze odată ajunse în cadrul contractului, până să ajungă acolo este nevoie de o tranzacție care să le trimită, tranzacție publică. Astfel, adăugarea de date senzitive pe blockchain necesită un pas suplimentar pentru a ne asigura că păstrăm în siguranță datele senzitive ale utilizatorilor.

2.9. Posibile soluții

În cadrul acestui subcapitol voi evidenția câteva soluții pe care le-am utilizat în aplicația **Medchain**.

2.9.1. Stocarea de fișiere – IPFS

În ceea ce privește stocarea persistentă de fișiere mari (spre exemplu fișiere de tip PDF, etc.) nu putem să ne rezumăm la capabilitățile blockchain-ului, întrucât acesta se pliază mai mult pe tranzacții de dimensiuni mici și rapide, cu un conținut redus de informații (cine, unde trimite, ce apelează și cu ce date de intrare apelează anumite bucăți de cod din contracte).

O soluție la această problemă se poate regăsi într-un protocol precum IPFS („InterPlanetary File System"). IPFS are ca aspirație să înlocuiască protocolul HTTP, fiind la rândul lui un protocol hypermedia peer-to-peer, observând că este mult mai eficient să descarci conținut dintr-o astfel de rețea distribuită, față de abordarea clasică prin care ne conectăm la un server ce ne oferă respectivul conținut spre a fi descărcat.

IPFS funcționează în felul următor: (4)

- Fiecare fișier ce urmează să fie încărcat pe rețea și blocurile din cadrul acestuia (bucăți de bytes din respectivul fișier) primește o semnătura digitală unică (se reîntâlnește conceptul de hashing).
- IPFS șterge orice fișier duplicat de pe rețea, informația există o singură dată distribuită pe mai multe noduri din rețea.
- Fiecare nod din rețea păstrează conținutul de interes și metadate legate de ce noduri cunoaște, ce fișiere au și unde se află acestea.
- Când cauți un fișier, practic ceri rețelei să-ți ofere nodurile care păstrează conținutul care te interesează în funcție de hash-ul generat al acestuia.

Astfel, ne putem folosi de IPFS pentru a stoca în mod persistent date de dimensiuni considerabile, având acces, apoi, la un hash unic ce ne indică unde se află respectivul fișier pe rețeaua IPFS. În cadrul unei tranzacții pe blockchain, pentru a include un fișier într-o tranzacție vom insera doar semnătura digitală a acestuia ce indică locația fișierului pe IPFS, astfel reducând considerabil dimensiunea unei tranzacții.

Un element și mai interesant este faptul că putem adăuga și găzdui în cadrul IPFS și pagini web sau aplicații client pentru a servi utilizatorii, acestea putând fii accesate folosind nume ușor de reținut furnizate de un sistem DNS descentralizat oferit de organizația IPFS, prescurtat IPNS („InterPlanetary Name System”).

2.9.2. Intimitate – criptarea datelor senzitive

Una din problemele de care ne lovim atunci când dezvoltăm aplicații pe blockchain, în particular Ethereum, este expunerea de date personale și senzitive ale utilizatorilor într-un mod public în istoricul blockchain-ului. Deși acesta are ca beneficiu transparența, simplul fapt că putem expune cu ușurință datele utilizatorilor în istoricul blockchain-ului este o problemă serioasă de fezabilitate a tehnologiei.

Deși Ethereum furnizează mecanisme și metode prin care poți cripta datele în cadrul unui contract, nu furnizează și mecanisme prin care conținutul trimis către contract să fie criptat. Astfel, orice date trimise la apelul unei funcții a unui contract sunt publice în istoric sub formă de „bonuri” ale tranzacțiilor (răspunsuri legate de reușita validării sau invalidării tranzacțiilor).

Figura 5 - Exemplu de apel al unei funcții din cadrul unui contract

Așa cum observăm în Figura 5, datele senzitive, precum coduri numerice personale, sau orice alte date ce pot fi trimise către un contract spre a-și executa logica, sunt publice.

Într-o astfel de situație ne putem folosi de diverse moduri de criptare a datelor înainte ca acestea să fie trimise către contracte și infrastructura Ethereum. Spre exemplu, s-ar putea folosi criptarea simetrică (cu o singură cheie secretă) pentru a trimite și a stoca astfel datele, ca ulterior

pentru accesare și decodificare să fie nevoie de respectiva cheie spre a decripta datele. Bineînțeles, se pot folosi diverse abordări, dar ideea principală este că există necesitatea implementării sau utilizării unui mecanism de criptare separat de ce poate oferi Ethereum, daca aplicația are nevoie de păstrarea în siguranță a datelor senzitive ale utilizatorilor.

3. Aspecte practice

În acest capitol voi expune aspectele practice din cadrul proiectului dezvoltat. Voi descrie ce funcționalități mi-am propus să aibă aplicația, cum arată și cum a fost implementată arhitectura acesteia, respectiv voi intra în detalii în ceea ce privește componentele aplicației.

3.1. Medchain – privire de ansamblu

Aplicația „**Medchain**” își propune să rezolve o problemă concretă în sistemul medical autohton, dar și la nivel global și anume ușurința cu care istoricul medical poate să fie pierdut în diferite situații, catastrofe ori în mânuirea lor fără grijă, istoric medical ce oferă un bagaj de informații util în luarea deciziilor pe termen scurt, mediu și lung legat de sănătatea pacienților. De asemenea, o altă problemă întâlnită, legat de istoricul medical, este fragmentarea acestuia în diverse baze de date, dar fiind în continuare păstrate într-un mod centralizat, fie în format digital, fie chiar în format fizic.

O serie de funcționalități pe care aplicația le implementează:

- Oferirea posibilității țărilor de a fi înscrise în aplicație și de a le permite să introducă în sistem entități medicale (indivizi, spitale publice, spitale private, etc.) ce își desfășoară activitatea pe teritoriul țării respective.
- Validarea țărilor înscrise printr-un cont de autoritate supremă (ce poate să fie o țară participantă în cadrul aplicației, deja înregistrată).
- Pacienții se pot înscrie în aplicație pe baza codurilor unice personale (spre exemplu, se pot înregistra folosind o combinație a CNP-ului și codului unic de asigurat medical) și furnizarea unei entități medicale ce urmează să valideze intrarea pacientului în sistem.
- Entitățile medicale pot, de asemenea, să se înscrie singure sub tutelajul unei țări, având nevoie de aprobarea acesteia pentru a avea un cont activ.
- Pacienții pot să își vizualizeze istoricul medical în diferite moduri (în funcție de tipul istoricului medical, de aria medicală, în funcție de entitatea medicală la care a apelat, etc.). De asemenea, aceștia vor putea vizualiza lanțuri de înregistrări medicale (înregistrări ce se leagă între ele, spre exemplu: vizită medicală de rutină – diagnostic – tratament – vizită medicală finală).

- Istoricul medical al pacienților este privat și poate fi vizualizat, în primă fază, doar de către acesta. Există, însă, o categorie numită „Sumar de Urgențe” ce poate fi vizualizat de orice entitate medicală, chiar dacă nu este asociată cu respectivul pacient, cât timp dispune de codurile de identificare personală ale respectivului pacient. Această funcționalitate este utilă în situații de urgență când este necesară vizualizarea rapidă a datelor de interes de către orice entitate medicală ce este responsabilă, în momentul respectiv, de urgență.
- Entitățile medicale înregistrate în sistem pot citi istoricul medical al pacienților cu care au interacționat doar dacă dispun de codurile personale cu care aceștia s-au înregistrat. Daca un pacient nu a interacționat vreodată cu o entitate medicală, aceasta nu poate să citească istoricul pacientului fără a cere asocierea cu el și aprobarea din partea pacientului.
- Entitățile medicale pot scrie în istoricul pacienților cu care sunt asociați doar dacă dispun de codurile personale de identificare ale acestora. Scrierea în istoric dispune de un mecanism prin care pacienții pot vizualiza ce urmează să fie scris în istoricul lor. Astfel, prima dată când o înregistrate medicală este trimisă către pacient spre a fi scrisă în istoricul acestuia, aceasta ajunge în secțiunea de „Înregistrări în așteptare”, loc unde pacientul poate să se asigure că înregistrarea este corectă și o poate aproba spre a fi înregistrată definitiv, ori trimisă înapoi către entitatea medicală spre a fi corectată. În situația din urmă, în urma eventualelor corectări, respectiva înregistrare medicală va fi scrisă direct în istoricul pacientului.
- În cadrul înregistrărilor medicale se pot atașa fișiere, arhive și alte date importante ce pot constitui informații de interes pentru pacienți și alte entități medicale, fișiere ce sunt stocate în cadrul IPFS.

3.2. Arhitectura aplicației Medchain

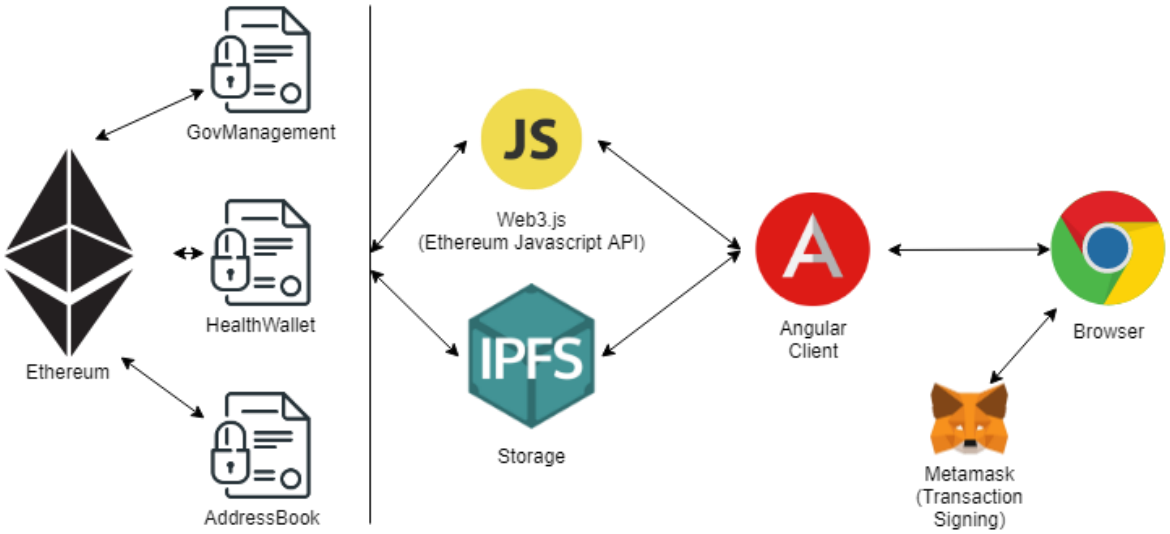

Figura 6 - Arhitectura aplicației Medchain (5)

În ceea ce privește arhitectura aplicației **Medchain**, avem în prim-plan infrastructura oferită de Ethereum în cadrul căreia am dezvoltat trei contracte, fiecare având un scop concret. Ce este important de menționat în această zonă este faptul că unele contracte au o oarecare dependență legată de celelalte contracte. Spre exemplu, cel mai mare contract, HealthWallet, are nevoie de instanța contractului AddressBook cu scopul de a verifica anumite legături între pacienți și entitățile medicale.

Comunicarea dintre contracte și punctul final, browser-ul, se face printr-un API furnizat de dezvoltatorii Ethereum, librărie ce se intitulează **Web3.js**. Aceasta expune o serie de metode ce facilitează operații precum extragerea instanțelor de contracte de pe blockchain, crearea și trimiterea de tranzacții către blockchain, o serie de unelte prin care diferitele tipuri de excepții și erori ce pot apărea în urma tranzacțiilor pot fi prinse și păstrarea unei conexiuni cu Ethereum, printre altele.

Pe partea de client avem o aplicație scrisă în ultima versiune de Angular la momentul actual, ce se folosește de librăria Web3.js alături de o conexiune deschisă către rețeaua oferită de IPFS pentru stocare persistentă atunci când este nevoie. În principiu, atunci când utilizatorul dorește să întreprindă o acțiune în aplicație ce necesită stocarea de fișiere, această tranzacție este formată la nivel de client, se așteaptă stocarea în cadrul IPFS de unde se obține un hash către respectivul fișier, se adaugă hash-ul la datele de intrare ale tranzacției și cu ajutorul Web3.js se

face trimiterea acesteia către contractele de pe Ethereum, de unde logica scrisă în contracte este executată și operația este finalizată, primindu-se un răspuns, sau nu, în funcție de operație.

Important de menționat este o extensie de browser, Metamask, folosită cu scopul de a semna tranzacții. Așa cum am specificat în capitolele anterioare, conturile de utilizatori din cadrul oricărui blockchain sunt definite printr-o pereche de chei, una publică și una privată ce te identifică în mod unic pe platformă și îți permit să faci operații cât timp le ai pe ambele asupra ta. Pentru a face managementul acesta de adrese (conturi de utilizatori) mă folosesc de un astfel de „portofel Ethereum". De asemenea, permite accesul în cadrul aplicațiilor descentralizate fără a fi necesar să rulezi propriul nod Ethereum.

3.3. Contractele aplicației

În cadrul acestui subcapitol voi expune pe larg responsabilitățile fiecărui contract dezvoltat în aplicația **Medchain**.

3.3.1. GovManagement

Acest contract este responsabil cu managementul de țări și entități medicale. Prima dată când acest contract este încărcat pe Ethereum va înregistra adresa celui care a făcut această tranzacție și va fi considerată drept autoritatea aplicației. Nevoia unei forme de autorități survine din faptul că, daca nu ar exista, orice individ s-ar putea înregistra în mod gratuit drept o țară a lumii, motiv pentru care credibilitatea aplicației ar scădea, neputând fii sigur daca respectivele adrese sunt sau nu ale unor guverne reale. Cât timp ele sunt validate de o autoritate, credibilitatea crește, o autoritate putând fii chiar o țară care să dea startul acestei aplicații pe Ethereum.

Dincolo de partea de management al guvernelor, odată înregistrate în sistem, acestea pot adăuga entități medicale ce își desfășoară activitatea pe teritoriul țării respective, ori să accepte sau refuze cereri de înscriere din partea entităților medicale ce încearcă să se înregistreze în respectiva țară. De asemenea, guvernele pot să suspende activitatea unei entități medicale, dar nu o pot șterge din sistem.

3.3.2. AddressBook

Acest contract este responsabil cu ținutul evidenței a cine cu cine a interacționat în cadrul aplicației și al înregistrărilor de pacienți pe platformă. Spre exemplu, daca un pacient s-a

înscris, implicit a fost nevoit să specifice o adresă a unei entități medicale pentru a putea fi validat contul. Astfel de interacțiuni sunt păstrate la nivel de contract, ca ulterior să aibă acces la o listă de contacte legată de entitățile medicale cu care a interacționat în diverse situații.

Facilitează translatarea adreselor prin care conturile se identifică pe Ethereum într-o formă citibilă de către oameni. Același lucru este valabil și pentru entitățile medicale care, cu ajutorul acestui contract, au acces la pacienții pe care îi are asignați, poate să observe ce cereri noi de pacienți care doresc să se asigneze la respectiva entitate mai sunt și poate să aleagă să renunțe la anumiți pacienți.

3.3.3. HealthWallet

Acest contract este cel care se ocupă de managementul și stocarea istoriilor medicale ale pacienților. Se face o despărțire, pe lângă categoriile clasice de înregistrări medicale, între înregistrări normale și cele ce intră în Sumarul de Urgență al pacientului, sumar ce conține informații și intrări de sistem esențiale în situații limită, acesta existând cu scopul de a putea fi citit și de entități medicale ce încă nu au asignați respectivii pacienți, de regulă, după cum spune și numele, pentru situații de urgențe.

De asemenea, se urmăresc două tipuri de înregistrări. Cele clasice și cele care conțin o formă de diagnostic. Cele din urmă sunt esențiale pentru a filtra rapid istoricul cu scopul de a avea o privire de ansamblu asupra maladiilor pe care pacientul le-a suferit în decursul vieții.

Un alt mecanism implementat în cadrul contractului este cel de cozi de așteptare pentru înregistrările medicale. Odată ce entitatea medicală dorește să scrie în istoricul unui pacient, această înregistrare ajunge inițial în coada de așteptare a acestuia, loc de unde pacientul poate să analizeze respectiva înregistrare și se poate asigura că tot ceea ce i-a fost expus, eventual prin viu grai, se translatează în format digital. De aici el poate să accepte sau să refuze incluziunea înregistrării în istoricul său, necesitând oferirea unui motiv. În cazul refuzului și completării motivului, înregistrarea ajunge înapoi la entitatea medicală ce poate să modifice anumite câmpuri ale înregistrări, ca de data aceasta să fie inserată direct în istoricul pacientului, sărind peste pasul legat de cozi. Acest mecanism permite diminuarea de posibile greșeli (precum încărcarea de fișiere greșite pentru respectiva înregistrare) ce urmează să rămână permanent în istoricul pacientului.

Pe lângă acestea, contractul oferă acces la datele pacienților doar entităților ce au dreptul de a vizualiza datele respective.

3.4. Legătura dintre contracte și aplicația client

Așa cum am menționat și în subcapitolul legat de arhitectura aplicației, legătura dintre client (scris în Angular) și platforma Ethereum se face prin API-ul furnizat de organizația Ethereum. Totodată ne folosim de extensia Metamask pentru a comunica cu rețeaua Ethereum și a ne autentifica în cadrul aplicației.

La prima intrare în aplicație, daca nu am mai interacționat cu ea vreodată de pe contul pe care intrăm, vom fi întâmpinați de un dialog din partea Metamask ce ne întreabă dacă suntem de acord ca al nostru cont să fie înregistrat și să-l putem folosi în această aplicație descentralizată.

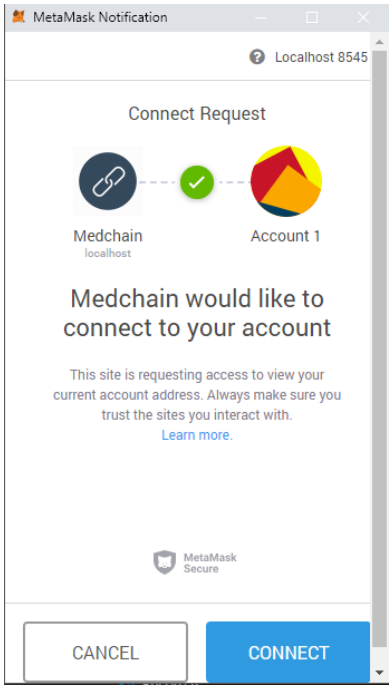

Figura 7 - Metamask cere confirmarea pentru conectarea pe Medchain

Deși aplicație e gândită să poată funcționa și fără un intermediar, această extensie oferă o serie de facilități utile, precum păstrarea în siguranță a conturilor, mecanisme de securitate împotriva aplicațiilor malițioase și un mod ușor prin care să-ți vizualizezi soldul conturilor.

La nivel de cod, după cum spuneam, aplicația e gândită să funcționeze fără Metamask

```
private bootstrapWeb3() {
    window.addEventListener('load', async () => {
        if (window.ethereum) {
            this.web3Object = new Web3(window.ethereum);

            try {
                await window.ethereum.enable();
            } catch (error) {
                console.log(error);
            }
        }
        else if (window.web3) {
            this.web3Object = new Web3(window.web3.currentProvider);
        }
        else {
            this.web3Object = new Web3(new Web3.providers.HttpProvider('http://localhost:8545'))
        }

        setInterval(() => this.refreshAccounts(), 100);
        return this.web3Object;
    })
}
```

Figura 8 - Configurarea Web3.js

Mai concret, daca nu există proprietatea „ethereum” în cadrul paginii noi încărcate, implicit nu există Metamask instalat în browser, atunci va încerca fiecare variantă, de la a crea un obiect de tip web3 dintr-un furnizor din pagina curentă (o altă extensie, alta decât Metamask, ce poate să ofere un endpoint către Ethereum), iar în ultimă instanță se apelează la crearea unei conexiuni către un endpoint specific (în cadrul aplicației s-a folosit un nod local Ethereum, dar la fel de bine se pot folosi diferiți furnizori ce oferă acces către Ethereum într-un mod simplu și rapid).

De asemenea, datele care sunt trimise în cadrul tranzacțiilor sunt inițial criptate folosind un algoritm de criptare simetric, Blowfish (6). Am ales acest algoritm de criptare întrucât este un algoritm foarte rapid și în același timp consumă destul de puține resurse la nivel de browser, elemente importante, ținând cont că toate datele tranzacțiilor trebuie criptate și toate istoriile medicale aduse de pe stocarea contractelor trebuie să fie decriptate spre a fi inteligibile de către oameni.

Concluziile lucrării

În urma dezvoltării proiectului am reușit să aprofundez mai în detaliu conceptul de blockchain și tehnologia din spate, cum funcționează și cât de utilă este pentru viitorul de mâine al Internetului și al aplicațiilor acestuia. Am reușit să pornesc de la o idee simplă la o implementare concretă, utilizând tehnologii și limbaje noi și o varietate de noi abordări în materie de a scrie cod (precum cod orientat pe contracte). Astfel, am putut translata niște concepte definite de platforma Ethereum într-un produs concret și util și consider că a fost o experiență plăcută și extrem de benefică pentru evoluția mea ca dezvoltator.

O posibilă îmbunătățire a aplicației ar fi o integrare cu dispozitive inteligente ce salvează diverși indici și statistici de sănătate ale utilizatorului în timp real. Din cadrul acestor date se pot observa tendințe și posibile apariții ale problemelor de sănătate încă de la început. Alături de aceasta, s-ar putea defini un mecanism de predicție a unor boli, atenționări ori fel de fel de interacțiuni cu utilizatorul acestor dispozitive. Datele ar fi stocate folosind implementarea curentă, posibil cu o abordare mai rapidă, întrucât vorbim de o cantitate mare de date ce pot fi trimise continuu.

O altă îmbunătățire ar fi dezvoltarea unui contract / modul în cadrul aplicației care să furnizeze unelte prin care pacienții să poată să-și ofere, fie voluntar, fie într-un mod plătit, istoriile medicale cu scopul de a forma „research pool-uri”. Țările interesate ar putea utiliza așa ceva pentru instituții de cercetare, ori diverse studii ce trebuie făcute pe eșantioane de oameni, loc unde există nevoie de voluntari.

O altă îmbunătățire ar fi extinderea contractului GovManagement pentru a furniza unelte și statistici, altele decât cele prezente acum în aplicație, pentru țări, în ceea ce privește managementul entităților medicale. Posibil ca în viitor, entitățile medicale ar putea fi plătite direct dintr-o astfel de aplicație, într-un mod simplu, transparent și rapid.

Bibliografie

1. http://adilmoujahid.com/posts/2018/03/intro-blockchain-bitcoin-python/ - **imagine cu problema Double-Spending**

2. https://en.wikipedia.org/wiki/Merkle_tree - imagine cu Merkle Tree

3.https://medium.com/@fhansmann/smart-contracts-and-the-real-world-a-complicated-relationship-c00dd766731a - imagine cu vizualizarea contractelor în Ethereum

4. https://ipfs.io/

5. https://www.draw.io/ - platformă de unde am generat imaginea cu arhitectura

6. https://github.com/agorlov/javascript-blowfish - implementare a algoritmului Blowfish

7. Blockchain Fundamentals. https://msdn.microsoft.com/en-us/magazine/mt845650.aspx

8. What is a Merkle Tree. https://blockonomi.com/merkle-tree/

9. Buterin, Vitalik. The Meaning of Decentralization. https://medium.com/@VitalikButerin/the-meaning-of-decentralization-a0c92b76a274

10. Top five blockchain benefits transforming your industry. https://www.ibm.com/blogs/blockchain/2018/02/top-five-blockchain-benefits-transforming-your-industry/

11. Ethereum Wiki https://github.com/ethereum/wiki/wiki

12. Ethereum Reading List For Prospective Dapp Developers. https://dappdaily.com/ethereum-reading-list-for-prospective-dapp-developers-15d515383b23