

UNIVERSITATEA “ALEXANDRU IOAN CUZA” DIN IAȘI

FACULTATEA DE INFORMATICĂ



LUCRARE DE LICENȚĂ

Încredere și Reputație în Rețele Sociale

Propusă de:

Onuțu Codrin-Ștefan

Sesiunea: *iulie, 2017*

Coordonator Științific
Conf. Dr. Alboaie Lenuța

UNIVERSITATEA “ALEXANDRU IOAN CUZA” DIN IAȘI

FACULTATEA DE INFORMATICĂ

Încredere și Reputație în Rețele Sociale

Onuțu Codrin-Ștefan

Sesiunea: *iulie, 2017*

Coordonator Științific
Conf. Dr. Lenuța Alboaie

DECLARAȚIE PRIVIND ORIGINALITATE ȘI RESPECTAREA DREPTURILOR DE AUTOR

Prin prezenta declar că Lucrarea de licență cu titlul „Încredere și Reputație în Rețele Sociale” este scrisă de mine și nu a mai fost prezentată niciodată la o altă facultate sau instituție de învățământ superior din țară sau din străinătate. De asemenea, declar că toate sursele utilizate, inclusiv cele preluate de pe Internet, sunt indicate în lucrare, cu respectarea regulilor de evitare a plagiatului:

- toate fragmentele de text reproduse exact, chiar și în traducere proprie din altă limbă, sunt scrise între ghilimele și dețin referința precisă a sursei;
- reformularea în cuvinte proprii a textelor scrise de către alți autori deține referința precisă;
- codul sursă, imaginile etc. preluate din proiecte open-source sau alte surse sunt utilizate cu respectarea drepturilor de autor și dețin referințe precise;
- rezumarea ideilor altor autori precizează referința precisă la textul original.

Iași, iulie 2017

Absolvent

Onuțu Codrin-Ștefan

Declarație de consimțământ

Prin prezenta declar că sunt de acord ca Lucrarea de licență cu titlul „Încredere și Reputație în Rețele Sociale”, 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, iulie 2017

Absolvent

Onuțu Codrin-Ștefan

Cuprins

1	Contribuții	8
2	Contextul actual	9
3	Rețele Sociale	11
3.1	Scurt Istoric	11
3.2	Rețele Profesionale	12
3.3	Concluzii	13
4	Încredere și Reputație	14
4.1	Încredere și Reputație în comunități online	14
4.2	StarTrust	15
4.2.1	Componenta pentru determinarea încrederii	16
4.2.2	Recomandarea resurselor	17
4.2.3	Componenta de calcul a reputației	17
5	Modelul Propus	18
5.1	Experiența Utilizatorului	18
5.2	Modelarea datelor	19
5.3	Încrederea	19
5.3.1	Resurse	19
5.3.2	Utilizatori	21
5.3.3	Reputația	22
5.4	Concluzii	22
6	Implementarea	23
6.1	Proprietățile sistemului	23
6.1.1	Backend	23
6.1.2	Clientul	25
6.2	Implementarea autorului	26
6.2.1	Serviciul	26
6.2.2	Clientul	28

Introducere

În prezent, trăim într-o comunitate ce este profund conectată prin intermediul rețelelor media sociale. Împreună cu rețelele media sociale obișnuite au apărut și rețele media profesionale, ce au ca scop facilitarea comunicării dintre utilizatori în scopuri profesionale, aceștia având la îndemână metode de a se autopromova.

Pentru aceste tipuri de rețele media este conceput și sistemul prezentat în această lucrare. Acesta are ca scop optimizarea conținutului pentru utilizatorii unor astfel de rețele prin intermediul conceptelor de încredere și reputație ale utilizatorilor, ajutând la crearea unei experiențe optime pe platformele de acest tip.

Prezenta lucrare va descrie modul în care s-a creat un astfel de sistem de măsurare a încrederii și reputației. Pornind de la un model deja existent, acesta a fost modificat pentru a fi compatibil cu reprezentarea specifică a componentelor unei platforme profesionale. De asemenea, se va prezenta și o posibilă implementare a unui astfel de sistem, făcută de autor pentru a demonstra beneficiile optimizării de conținut pentru utilizatori față de organizarea informației după alte criterii (precum ordinea cronologică).

Soluția presupune modelarea resurselor (umane sau nu) în două categorii separate: utilizatori și resurse. Resursele vor reprezenta, în cazul nostru postări ale utilizatorilor în cadrul rețelei media sociale, fie ele actualizări de stare sau oferte pentru locuri de muncă oferite de către companii. Pentru calcularea nivelelor de încredere și a rating-urilor utilizatorilor se vor folosi o suită de tehnologii ce se află atât în componenta de back-end a aplicației cât și în aplicația client pentru o astfel de rețea.

Am ales această temă din dorința de crea un model de calcul al încrederii și reputației ce răspunde nevoilor unei rețele media profesionale, modelul fiind adaptat particularităților comunităților de acest tip. Am dorit să creez un model care să vină în ajutorul utilizatorului fără a îi împovăra experiența sa de navigare, optimizând în același timp conținutul pe care îl vede. Un alt aspect ce m-a atras asupra acestei teme a fost provocarea de a proiecta un sistem care să analizeze cantități uriase de date în funcție de nevoile utilizatorului, oferindu-i resursele disponibile pe platformă într-un mod personalizat după nevoile și dorințele sale.

Deși problema încrederii și a reputației în rețele sociale nu este una nouă în domeniu, soluția propusă construiește folosind rezultate recente în domeniu în speranța obținerii unui sistem fezabil pentru rezolvarea problemelor de acest tip în cadrul rețelelor profesionale.

1 Contribuții

În cadrul acestui proiect am avut contribuții pe următoarele planuri:

- Pe plan teoretic:
 - Definirea modelului de calcul a încrederii, folosind ca punct de plecare StarTrust. Acesta a presupus analiza modului de interacționare dintre utilizator și conținut, crearea unui model pentru conversie în interacțiunea utilizator-resursă.
 - Definirea modurilor de calcul al încrederii, având în vedere modul în care utilizatorii interacționează unul cu celălalt.
- Pe plan practic:
 - Identificarea cerințelor implementării unei astfel de aplicații.
 - Investigarea posibilelor soluții de implementare pentru modelul propus.
 - Implementarea modelului folosind baza de date Neo4j, împreună cu scripturile Cypher necesare pentru a realiza modelul.
 - Implementarea aplicației client pentru a arăta un model de captare a informațiilor de la client pentru calcularea încrederii, și, implicit, a reputației.

2 Contextul actual

În prezent, trăim într-o comunitate ce produce foarte multe date în mediul online. Spre exemplu, în 2014, compania Facebook avea o bază de date de circa 300 de petabytes ¹, cu o rată de ingestie de circa 600 de terabytes pe zi. Odată cu apariția terminalelor mobile cu acces la Internet și ieftinirea dispozitivelor ce se pot conecta la acesta, cantitatea de date produsă de către oameni a crescut exponențial. Am ajuns astfel să fim copleșiți de volumul de date pe care îl creăm.

Ca și răspuns la această problemă a apărut conceptul de Big Data(cantități mari de date), un domeniu al informaticii ce se preocupă strict cu procesarea volumelor mari de informație. Odată cu acest curent, a apărut și noțiunea de Big Data Analytics(analize pe cantități mari de date), ce a dus companiile cu platforme ce produceau cantități mari de date să privească datele nu doar ca locuri pentru a stoca informații despre performanțele sistemului, ci locuri pentru a dobândi informații despre clienții lor, în special modul lor de navigare în cadrul paginilor lor.

Astfel, a apărut un trend de optimizare a conținutului cu afișare către utilizator care să îl conducă pe acesta să îndeplinească activități benefice proprietarului platformei.

În urma acestui trend a apărut și nevoia de a găsi moduri de a ajuta utilizatorul să găsească mai ușor conținutul dorit. De asemenea conținutul acesta putea fi personalizat astfel încât utilizatorul să primească rezultatele dorite conform dorințelor sale, dar și care aduceau profitabilitate maximă deținătorului platformei.

Astfel, companiile au început să implementeze module a căror simplu scop era să facă recomandări cât mai relevante utilizatorilor. Acesta a fost un trend destul de major între platformele de comerț electronic, întrucât un utilizator putea mereu să treacă pe platforma unui competitor pentru simplul fapt că avea o soluție de recomandare mult mai bună.

Studiu de caz: BelVilla

O astfel de companie care a implementat un sistem de recomandare a fost compania BelVilla. Aceasta operează un site dedicat închirierii caselor de vacanță. Întrucât clienții intră foarte rar pe platformele de acest tip, aceștia aveau nevoie să se asigure că clienții se vor întoarce pe platformă după un timp.

După o analiză a traficului, aceștia au observat că foarte puțini clienți se întorceau. Era așadar nevoie de o rată de conversie mult mai mare a clienților, întrucât ei abandonau căutările la un timp foarte scurt, fiind nemulțumiți de rezultate.

¹<https://code.facebook.com/posts/229861827208629/scaling-the-facebook-data-warehouse-to-300-pb/>

Astfel, au construit o soluție de recomandare², reușind să îmbunătățească rata de satisfacție a clienților. O altă oportunitate pe care au găsit-o sistemului a fost în relația cu clienții. Astel, în momentul în care erau probleme cu rezervările unui client, operatorii puteau imediat să acceseze sistemul și să afle o proprietate recomandată pentru clientul în dificultăți.

Sistemul lor este bazat pe analiza unui graf format din milioane de noduri ce reprezintă atât proprietăți cât și trăsături ale lor, acestea fiind procesate pentru a realiza recomandările către clienți.

²<https://www.youtube.com/watch?v=7sMctFYXW8U>

3 Rețele Sociale

Înca de la începutul anilor 90 s-a cunoscut o creștere majoră în rata de acces la Internet și a deținătorilor de terminale cu acces la Internet. Astfel, la sfârșitul lui 2016, 43,83% din populația lumii folosea Internetul.

Rețelele sociale au devenit o parte a rutinei zilnice pentru milioanele de utilizatori pe care i-au atras. În prezent, există sute de rețele de socializare, fiecare cu particularitățile ei, permițând utilizatorilor să fie conectați unul cu celălalt după diverse criterii caracteristice rețelei de socializare. În mod formal, „rețelele media sociale sunt servicii web ce permit indivizilor să: (1) construiască un profil public sau parțial-public în cadrul unui sistem limitat, (2) să realizeze o listă de utilizatori cu care împărtășesc o conexiune și (3) să vadă și să parcurgă lista lor de conexiuni, cât și listele create de alți utilizatori ai sistemului. Natura și nomenclatura acestor conexiuni pot varia de la sistem la sistem”.[1]

Componentele de bază ale unei rețele sociale sunt lista de conexiuni, respectiv o pagină de profil în care se pun la dispoziția altor utilizatori informații în legătură cu individul ce deține profilul. În general acest profil este completat la momentul înregistrării, prin intermediul unui formular ce conține câmpuri precum “Vârstă”, “Orașul natal”, dar și o secțiune “Despre” ce permite utilizatorului să se descrie folosind propriile cuvinte.

3.1 Scurt Istoric

Prima rețea socială a apărut în 1997, aceasta fiind SixDegrees.com. SixDegrees avea toate componentele de bază ale unei rețele sociale. Din nefericire, în ciuda faptului că a reușit să atragă un număr considerabil de utilizatori(câteva milioane), aceasta s-a închis în anul 2000, într-un moment în care rețelele media sociale încă nu luaseră amploare.

Un următor val de rețele a apărut odata cu Ryze.com în 2001. Spre deosebire de SixDegrees, Ryze era menit strict pentru rețelele profesionale. În același timp aveau să apară LinkedIn și Friendster. Friendster a fost un succes aproape instant, dar, din păcate, a fost curînd depășit de MySpace și, ulterior Facebook. LinkedIn a devenit una dintre cele mai populare rețele sociale profesionale.

În 2004 a luat naștere Facebook-ul. Inițial conceput ca o rețea de socializare pentru studenți acesta a devenit curînd un fenomen. Astfel, aproximativ 13 ani mai târziu, în Martie 2017, Facebook înregistra 1,94 miliarde utilizatori activi lunar, dintre care 1,28 miliarde se vizitau rețeaua de socializare zilnic.³

³<http://www.telegraph.co.uk/technology/2017/05/03/facebook-approaches-2-billion-users/>

Astfel, nu poate fi contestat faptul că pentru majoritatea persoanelor cu acces la Internet, activarea pe rețelele sociale este o activitate recurentă.

3.2 Rețele Profesionale

O categorie importantă a rețelor media sociale sunt cele profesionale. Ne vom axa pe acestea întrucât ele formează obiectul de studiu al prezentei lucrări.

Rețelele profesionale se diferențiază de celelalte prin stimularea creșterii unei rețele profesionale, comparativ cu rețelele sociale precum Facebook sau Twitter ce promovează împărtășirea de conținut legat de viața persoanlă. Astfel, platforme precum LinkedIn se remarcă prin intermediul promovării partajării de resurse profesionale. De asemenea, profilul nu mai este axat pe actualizări de orice natură, ci sunt concentrate strict pe viața profesională a deținătorului paginii. Astfel, deși procesul de înregistrare în rețeaua socială este similară, având preponderent aceleași câmpuri de bază, câmpuri precum “Despre” devin limitate, în sensul că utilizatorul este îndrumat în oferirea de răspunsuri prin prisma vieții sale profesionale.

Ca și studiu de caz vom considera compania LinkedIn, una dintre cele mai populare platforme de acest tip. Aceasta depășea în Aprilie 2017 pragul de 500 milioane utilizatori unici⁴. Acești membri proveneau din 200 de țări, aveau la dispoziție: peste 10 milioane de locuri de muncă, acces la peste 9 milioane de companii și peste 100 de mii de articole publicate săptămânal pe platformă. Depășirea acestui prag a făcut LinkedIn cea mai mare comunitate profesională ce a existat vreodată.

În cazul acesteia, conexiunile din rețea se numesc “conexiuni”, comparativ cu noțiunea de “prietenii” preponderent întâlnită în cazul altor rețele sociale. Astfel se sugerează utilizatorilor diferențele de scop ale platformei comparativ cu alte aplicații bazate pe comunități on-line. Lista de prietenii este numită “rețea”.

Diferența majoră este văzută în conținutul servit utilizatorilor platformei. Acesta este structurat în actualizări succinte sau articole. Actualizările sunt de obicei fie anunțuri în legătură cu oferte de muncă, fie actualizări ce împărtășesc opinii sau experiențe profesionale. Articolele în schimb sunt bazate pe împărtășirea de cunoștințe, resurse sau opinii într-o manieră mai formală decât actualizările de stare, având structura unui articol.

Comparativ cu alte platforme ce atrag prin posibilitățile de a împărtăși conținut media listei de conexiuni, aplicațiile de acest tip promovează conținutul textual, precum articolele. Acest fapt conduce la diferențe în modul de apreciere a conținutului comparativ cu alte comunități,

⁴<https://blog.linkedin.com/2017/april/24/the-power-of-linkedin-500-million-community>

fiind nevoie de soluții specifice acestor tip de rețele. Trebuie menționat că aceste platforme au totuși mecanisme familiare de apreciere ale conținutului, facilități precum posibilitatea de a oferi “aprecieri”(engl. likes), de a comenta asupra actualizărilor, actualizări ce pot fi și redistribuite.

3.3 Concluzii

Nu poate fi negat faptul că rețele sociale sunt printre cele mai populare tipuri de platforme pe Internet. Acestea atrag lunar sute de milioane de utilizatori unici ce aderă la comunități din mediul on-line împărtășindu-și gândurile și realizările în mediul virtual format de aceste platforme.

4 Încredere și Reputație

În interacțiunile umane încrederea și reputația au un rol extrem de important. Ele sunt folosite pentru a evalua în ce măsură putem evalua afirmațiile unei terțe ca fiind adevărate. Astfel, ne luăm deciziile pe baza unor cunoștințe sau aspecte evaluate anterior, decizia rezultată fiind una calculată, informată.

Putem considera încrederea și reputația ca fiind metrici comportamentale ce denotă probabilitatea ca un individ A să se bizuiească pe informația oferită de un alt individ B. Încrederea dintre A și B caracterizează relația lor. Dar gradul de încredere poate varia în funcție de actorul din perspectiva căruia vedem relația dintre A și B, întrucât A poate avea o părere (și, implicit, un grad de încredere) diferită față de părerea lui B asupra lui A. Așadar, în cazul în care am considera relația dintre A și B un graf, acesta ar fi un graf orientat, existând două arce diferite între A și B, de sens opus, cu valori diferite.

În timp ce încrederea sugerează o relație directă sau indirectă (încrederea poate fi moștenită de la un alt utilizator, după cum vom vedea ulterior), reputația este o metrică globală, rezultată ca o agregare opiniei colective asupra unui individ. Așadar, în elaborarea deciziilor, încrederea va prima mereu în fața reputației ca factor de luat în considerare, deoarece, chit că poate fi o încredere ce rezultă dintr-un lanț de încredere (spre exemplu A are încredere în B și B are încredere în C), încrederea este ponderată folosind mereu încrederea celui ce trebuie să ia decizia în comparație cu o decizie bazată pe reputație, unde ponderea adusă de individul ce trebuie să decidă este același cu al oricărui alt individ din colectivitate.

4.1 Încredere și Reputație în comunități online

În domeniul informaticii încrederea este adesea modelată ca o serie de probabilități, astfel încât încrederea este reprezentată prin valori numerice, acestea având următoarele proprietăți[2]:

- tranzitivitate: după cum am menționat mai sus, pot exista relații de tipul A are încredere în B, B în C, asta implicând ca A are încredere în C, aceasta fiind calculată folosind o tranzitivitate ce nu aderă sensului matematic întrucât parametrii se compun într-un mod diferit față de cel cunoscut, permițând formarea lanțurilor de încredere
- compozabilitate: abilitatea de a compune valori ale încrederii astfel încât ele să reprezinte mai multe opinii, în cazul în care A vrea să verifice mai multe opinii în privința lui C

- personalizarea: datorită multiplelor perspective asupra unui singur individ, calculele se efectuează începând cu perspectiva celui pentru care se calculează încrederea într-un alt utilizator, asigurând astfel că interesele sale sunt folosite în mod primar pentru calcularea încrederii.
- asimetria: cum am menționat, încrederea dintre A și B și cea dintre B și A pot varia semnificativ una față de cealaltă.

4.2 StarTrust

Ca și model de pornire pentru sistemul de încredere și reputație am ales modelul StartTrust[3]. În continuare vom descrie modelul astfel încât să putem analiza ulterior componentele ce aveau nevoie de modificări pentru a putea rula pe o rețea media profesională. StarTrust este un model compus din trei componente majore:

- Componenta de calculare a încrederii
- Componenta de recomandare a resurselor
- Componenta pentru calcularea reputației

Ca și notații, modelul utilizează următorii trei termeni:

- utilizatori - membrii comunității on-line
- resurse - acestea vor fi obiectele referite conform definiției lui Tim Berners-Lee în 1998, astfel, ele fiind orice entitate ce are identitate, fie un document, imagine serviciu sau o colecție de alte resurse.[4]
- valoare - reprezintă valoare returnată de o funcție de calcul a încrederii, asociată de un utilizator altuia. Aceasta simbolizează evaluarea acordată unui utilizator de către un altul. De asemenea, aceasta poate fi obținută indirect(și cuantificată) printr-un mod ce va fi detaliat ulterior. În StartTrust există 5 niveluri semantice pentru evaluare încrederii, după cum urmează:

Niveluri	Nivel 1	Nivel 2	Nivel 3	Nivel 4	Nivel 5
Interval	[0, 1]	(1, 2]	(2, 3]	(3, 4]	(4, 5]
Evaluare	nefolositor/spam	slab	merită interes	bun	excepțional

Tabela 1: Intervalele de încredere împreună cu evaluările asociate

Modelul notează valoarea maximă cu $MaxWorth$ unde, în model parametrul $MaxWorth$ are valoarea maximă 5.

4.2.1 Componenta pentru determinarea încrederii

Încrederea este calculată folosind o funcție ce returnează valoarea utilizatorului pentru mebrul din perspectiva căruia se face calculul. Din punct de vedere, modul de calculare poate fi considerat drept funcție sau drept tabele asociative.

Modelul calculează încredere pe baza a două scenarii, fiecare scenariu fiind caracterizat printr-o funcție separată. Aceste scenarii au la bază relația dintre utilizatorii pentru care se calculează încrederea. Astfel, pentru doi utilizatori A și B, determinarea scenariului se desfășoară astfel:

Cazul 1 În cazul în care A a evaluat vreodată pe B, se folosesc evaluările lui A pentru a determina valoarea explicită a lui B din punctul de vedere al lui A

Cazul 2 În cazul în care A nu a evaluat niciodată pe B, atunci se parcurge lista de utilizatori pentru care A deține o evaluare, fie ea directă sau indirectă, și pe baza evaluărilor directe pe care aceștia i le-au făcut lui B, se calculează valoarea implicită a lui B pentru A

Astfel, pentru primul scenariu s-a definit funcția $WE_{UU} : U \times U \rightarrow (0, 5]$ pentru încrederea explicită dintre utilizatori. Pentru încrederea implicită (Cazul 2) s-a definit funcția $WI_{UU} : U \times U \rightarrow (0, 5]$. Împreună, ele formează funcția pentru calculul încrederii din cadrul modelului pe care o definim astfel:

$$WU_{UU} : U \times U \rightarrow (0, 5]$$

$$WU_{UU}(U_1, U_2) = \begin{cases} WE_{UU}(U_1, U_2), & \text{dacă } U_1 \text{ a evaluat explicit pe } U_2 \\ WI_{UU}(U_1, U_2), & \text{altfel} \end{cases}$$

Având în vedere că WE_{UU} se bazează pe o evaluare directă a utilizatorului, rămâne de definit o relație care să caracterizeze WI_{UU} , care în modelul StartTrust este definită astfel:

$$WI_{UU}(U_i, U_j) = \frac{\sum_{l=1}^k WI(U_i, U_i^l) * WE(U_i^l, U_j)}{k * MaxWorth} + (1 - \frac{\sum_{l=1}^k WI(U_i, U_i^l)}{k * MaxWorth}) * P_{TC}(U_i, U_j)$$

Se poate observa prezența termenului P_{TC} , acesta având rolul unui parametru de control. În modelul StartTrust acesta poate varia în funcție de relație sau poate fi înlocuit cu reputația utilizatorului pentru care se calculează încrederea, în cazul în care în graful relațiilor de încredere nu există niciun drum între cei doi utilizatori.

4.2.2 Recomandarea resurselor

Într-un mod similar încrederii sunt caracterizate și relațiile utilizatorilor cu resursele disponibile pentru membrii comunității. Astfel, avem relațiile:

- $WE_UR : U \times R \rightarrow (0,5]$ - încrederea explicită a unui utilizator într-o resursă, obținută printr-o evaluare de către utilizator.
- $WI_UR : U \times R \rightarrow (0,5]$ - încrederea implicită a unui utilizator într-o resursă, obținută printr-o agregare a încrederii utilizatorilor ce fac parte din mulțimea utilizatorilor evaluați explicit de către utilizator
- $WR : U \times R \rightarrow (0,5]$ - încrederea unui utilizator într-o resursă

Mai exact, avem:

$$WR(U_i, R_j) = \begin{cases} WE_UR(U_i, R_j), & \text{dacă } U_i \text{ a evaluat explicit pe } R_j, \forall i \leq |U|, j \leq |R| \\ WI_UR(U_i, R_j), & \text{altfel} \end{cases}$$

WI.UR se calculează astfel:

$$WI_UR(U_i, R_j) = \frac{\sum_{l=1}^k WU(U_i, U_i^l) * WE_UR(U_i^l, R_j)}{k * MaxWorth}$$

4.2.3 Componenta de calcul a reputației

Conceptele de încredere și reputație sunt strâns corelate, astfel încât reputația unui utilizator poate fi percepută ca și o agregare a evaluărilor sale de către comunitate. Astfel, folosind intervalele de încredere precizate în Tabelul 11, se produce o agregare a rating-urilor sale după urmează. Fie un utilizator oarecare $U_i, \forall i \leq |U|$. Se notează $N_l^i = \{Ev_l^1, Ev_l^2, \dots, Ev_l^k\}$ mulțimea evaluărilor pe care U_i le-a primit în intervalul de încredere $l, 1 \leq l \leq MaxWorth$. Amintim că, în cazul StarTrust, $MaxWorth = 5$. Formula folosită în cadrul StarTrust este:

$$Rep(U_i) = \sum_{l=1}^{MaxWorth} \frac{|N_l^i|}{\sum_{j=1}^{MaxWorth} P_{LW}^j * |N_j^i|} * P_{RC}^l, \text{ unde:}$$

P_{RC}^l este parametrul de control al reputației

P_{LW}^j este gradul de semnificație al nivelului

5 Modelul Propus

În acest capitol vom detalia modelul propus pentru a implementa un astfel de sistem de încredere și reputație într-o rețea media profesională precum LinkedIn. Astfel, vom detalia reprezentarea utilizatorilor și a resurselor într-un astfel de mediu, cum ar putea fi calculată reputația și care ar fi cerințele pentru implementarea unui astfel de sistem. Vom analiza cum arată interacțiunile unui utilizator cu o astfel de platformă și cum ar trebui să se integreze un astfel de modul cu aplicația. Un exemplu de implementare va fi prezentat în capitolul următor.

5.1 Experiența Utilizatorului

Este evident că una dintre cei mai importanți factori pentru succesul unei rețele sociale este experiența utilizatorilor. Aceasta este responsabilă pentru atragerea clienților noi și păstrarea celor existenți. Așadar, pentru a câștiga și a păstra interesul utilizatorilor, experiența de navigare trebuie să fie un simplă și fluidă.

De asemenea, pentru a atrage multiple segmente demografice trebuie o experiență care să ofere cât mai multe beneficii, fără a îngreuna modul de interacționare cu platforma. Astfel, multe dintre serviciile platformei trebuie integrate în așa fel în platformă astfel încât să nu interfereze modului optim de navigare.

Pentru a putea analiza aceste date și a propune un model fezabil, trebuie studiat modul de interacționare cu aplicația al utilizatorilor. Acest lucru asigură că adăugarea unui modul de încredere și reputație nu perturbă modul optim de navigare în cadrul serviciului. De asemenea trebuie luată în considerare și modalitate de afișare a conținutului produs de acest modul, astfel încât să nu trebuiască refăcută arhitectura informației din cadrul platformei. Astfel, identificăm două caracteristici ale unui modul de acest tip ce sunt de o importanță majoră pentru succesul serviciilor furnizate de către acesta.

De asemenea, pentru a păstra experiența utilizatorului cât mai apropiată de cea optimă, din punct de vedere arhitectural acest modul trebuie să fie decuplat, pentru a permite aplicației să eșueze în mod tăcut, lipsa unui serviciu din suita oferită de platformă fiind preferată în locul unui eșec al întregii platforme.

Pentru a avea o experiență de navigare plăcută, serviciul trebuie să fie scalabil în funcție de necesități, asigurând livrarea conținutului către utilizatori într-un mod rapid, fiind de asemenea capabil să susțină creșteri de cereri odată cu evoluția rețelei. Din punct de vedere al deținătorului unui astfel de serviciu, acesta trebuie să fie scalabil orizontal, pentru a fi mai eficient din punct de vedere al costurilor.

Având în vedere cantitatea de date disponibilă pe o astfel de platformă, trebuie analizată experiența utilizator pe care este necesar să o furnizeze serviciul de încredere și reputație pentru a fi unul de succes. Rețelele sociale încurajează consumul a cât mai multe actualizări. Astfel, pentru a nu întrerupe calea normală de interacțiune cu platforma, un astfel de modul trebuie să extragă informațiile necesare calculului încrederii și reputației fără a cere informații suplimentare utilizatorului.

Din această cauză nu dorim să distragem atenția utilizatorului de la folosirea conținutului de bază a platformei(actualizări de stare, imagini, etc.) pentru a îi cere să evalueze conținutul pe care îl vizualizează.

5.2 Modelarea datelor

Pentru a putea beneficia de baza pusă de modelul StartTrust, trebuie găsită o modalitate de reprezenta actorii din cadrul platformei ca și tipurile de bază oferite de către acesta.

În mod natural, utilizatorii rețelei vor fi modelați ca și tipul omonim din cadrul StartTrust. Resursele vor fi toate actualizările platformei. Pentru a păstra compatibilitatea cu serviciile puse la dispoziție de către majoritatea platformelor, comentariile la resurse vor fi tratate la rândul lor ca și resurse, aprecierile date de utilizatori folosind butoanele similare celor de “like” de pe LinkedIn fiind reprezentate în cadrul modelului ca și etichete ale relației dintre utilizator și resursă. Asemănător, vom considera profilele utilizatorilor drept resurse, având totuși o altă legătură semantică față de cea pe care o are un utilizator cu o actualizare de stare.

5.3 Încrederea

După cum am menționat anterior, vrem ca încrederea să poată fi calculată fără a fi nevoie de a interveni în modul de navigare cu care utilizatorii platformelor profesionale sunt obișnuiți. Astfel, trebuie folosită o metodă cât mai puțin intrusivă de a utiliza acțiunile deja existente ca și parametri ai unei formule de calcul.

5.3.1 Resurse

Din acest considerent, este nevoie să definim încrederea în funcție de gradul de interacțiune al utilizatorului cu entitatea evaluată. Ca un prim pas vom defini încrederea pe care un utilizator o are într-o resursă. Întrucât pe o astfel de platformă, idealul unui utilizator ce crează conținut este ca resursa creată să fie cât mai apreciată și distribuită, astfel încât să ajungă la o audiență cât mai largă, care, în cazul ideal, să o aprecieze la rândul lor. În acest context, vom considera un flux

de acțiuni ce duc la rezultatul dorit. Evenimentul în care un utilizator ajunge la sfârșitul acestui flux de acțiuni manifestând comportamente ce arată aprecierea sa față de conținutul resursei și dorința de a o împărtăși va fi numit conversie.

Pentru a obține un model de măsurare al conversiei, trebuie analizată seria de pași pe care o îndeplinește un utilizator în evaluarea și distribuirea conținutului ce îi este prezentat pe o platformă socială. Primul pas, este întâlnirea resursei în cadrul fluxului de postări din cadrul aplicației. În mod natural, utilizatorul evaluează conținutul (prin citire, în cazul în care resursa este textuală, vizualizare în cazul imaginilor, deschiderea link-urilor, etc.). Aceasta este o primă interacțiune cu resursa. Ca și pași următori, utilizatorul poate să salveze postarea pentru a o citi mai târziu (în cazul în care platforma permite), să își manifeste aprecierea folosind butonul de “Like”, să comenteze asupra resursei sau să o distribuie mai departe listei sale de prieteni.

Formal, vom avea un drum al interacționării cu resursa compus din trei pași:

- 1 **Discovery** - acest pas presupune descoperirea evenimentului
- 2 **Interaction** - utilizatorul interacționează cu postarea, o citește, îi acordă interesul său
- 3 **Engagement** - implicarea utilizatorului cu resursa vizualizată, manifestată prin distribuiri, folosirea butonului pentru aprecieri, comentarii

Pentru a calcula încrederea utilizatorului în resursa prezentată, vom atribui un punctaj evenimentelor din fiecare categorie. Astfel, însumarea punctajelor atribuite acțiunilor realizate de către utilizator vor putea fi însumate pentru a obține nivelul de încredere în resursa respectivă. Vom atribui evenimentului de Discovery valoarea 0, întrucât descoperirea unui eveniment nu este la voia utilizatorului, ci a platformei ce servește conținutul. De asemenea, această mod permite la contorizarea resurselor față de care utilizatorul nu a prezentat interes, metrică pe care o putem folosi în a calcula încrederea medie în resursele publicate de către un utilizator, a cărei valoare o vom folosi în calcularea încrederii în utilizator.

În scopul formalizării acestui proces, vom considera următoarele funcții boolene care oferă diverse perspective asupra interacțiunii dintre utilizator și resursă:

- $hasDiscovered : U \times R \rightarrow \{0, 1\}$ - are valoarea 1 atunci când utilizatorului i-a fost randată resursa
- $hasRead : U \times R \rightarrow \{0, 1\}$
- $hasLiked : U \times R \rightarrow \{0, 1\}$
- $hasCommented : U \times R \rightarrow \{0, 1\}$

- $hasShared : U \times R \rightarrow \{0, 1\}$
- $hasBookmarked : U \times R \rightarrow \{0, 1\}$

Astfel, în cazul nostru, WE_UR va fi:

$$WE_UR(U_i, R_j) = hasRead(U_i, R_j) + hasLiked(U_i, R_j) + hasCommented(U_i, R_j) + hasShared(U_i, R_j) + hasBookmarked(U_i, R_j) \quad (1)$$

În timp ce WI_UR și WR_UR vor fi calculate asemănător modelului StartTrust.

5.3.2 Utilizatori

Interacțiunile între utilizatori continuă să joace un rol important în calcularea încrederii. Acestea semnifică interesul personal strict pentru un alt membru al comunității, fiind diferit de interesul asupra conținutului unei resurse.

Similar ne vom ajuta de predicate boolene pentru a modela evenimentele ce pot avea loc pe o platformă socială:

- $viewed : U \times U \rightarrow \{0, 1\}$ ce denotă dacă utilizatorul a vizualizat profilul celui pentru care se calculează încrederea
- $follows : U \times U \rightarrow \{0, 1\}$ dacă utilizatorul este abonat la postările celui evaluat
- $connected : U \times U \rightarrow \{0, 1\}$ dacă cei doi se regăsesc în listele de prieteni ai amândurora
- $endorsed : U \times U \rightarrow \{0, 1\}$ dacă cel evaluat a fost promovat public de către cel din perspectiva căruia se evaluează

Problematica ce se ridică este modul diferit de calcul al încrederii pentru utilizatori. În StartTrust, calculăm încrederea strict din prisma persoanei. În contextul rețelelor profesionale, unde persoanele nu se cunosc atât de bine din punct de vedere personal, intervine și evaluarea resurselor, întrucât persoanele se pot încrede în conținutul articolelor publicate, dar nu și în profilul persoanei. Astfel, trebuie inclusă și modul de interacționare cu resursele publicate în momentul determinării încrederii.

Așadar, WE_UU va fi calculat astfel:

$$R_{U_i} = \{r \in R \mid owns(r, U_i) = 1\}$$

$$WP_UU(U_i, U_j) = viewed(U_i, U_j) * 0,5 + follows(U_i, U_j) + connected(U_i, U_j) * 1,5 + endorsed(U_i, U_j) * 2$$

$$WE_UU(U_i, U_j) = \frac{\frac{\sum_{l=1}^k WE_UR(U_i, R_{U_i}^l)}{k * MaxWorth} + WP_UU(U_i, U_j)}{2 * MaxWorth}$$

Am notat R_{U_i} mulțimea resurselor deținute de U_i , și WP_UU rating-ul profilului determinat prin prisma acțiunilor utilizatorului.

Astfel, adaptăm calculul încrederii conform obiectivului platformelor de acest tip, în sensul că maximizăm beneficiul adus de natura concentrată pe conținut a aplicației.

5.3.3 Reputația

Modelul pentru reputație nu suferă modificări, întrucât aceasta rămâne o agregare a încrederii individuale în utilizator. Astfel, formula pe care o vom folosi rămâne aceeași, pe care o amintim:

Fie $N_l^i = \{Ev_l^1, Ev_l^2, \dots, Ev_l^k\}$ mulțimea evaluărilor pe care U_i le-a primit în intervalul de

încredere l , $1 \leq l \leq MaxWorth$ $Rep(U_i) = \sum_{l=1}^{MaxWorth} \frac{|N_l^i|}{\sum_{j=1}^{MaxWorth} P_{LW}^j * |N_j^i|} * P_{RC}^l$, unde:

P_{RC}^l este parametrul de control al reputației

P_{LW}^j este gradul de semnificație al nivelului

5.4 Concluzii

În acest capitol am discutat despre modul în care un sistem de încredere și reputație ar trebui să interacționeze cu platforma pentru a oferi utilizatorilor o experiență cât mai fluidă. Astfel, am realizat că un model precum cel descris în preenta lucrare trebuie să folosească modurile de evaluare a conținutului deja existente pe platformă pentru a extrage datele necesare calculului încrederii.

Am menționat de asemenea o schimbare în modul de lucru. Platformele sociale au ca scop principal consumarea conținutului produs de membrii comunității. Acest mod de lucru a dus la regândirea modului în care trebuie calculată încrederea într-un utilizator, întrucât un membru al comunității poate vedea pe un altul prin cel puțin două perspective: (1) prin prisma conținutului produs de acesta și (2) prin intermediul paginii sale de profil.

Astfel, modelul propus este construit pentru a veni în întâmpinarea particularităților unei astfel de platforme, ceea ce va duce la o acuratețe mai mare în calculul încrederii. O astfel de trăsătură a modelului este modelul pentru conversie, ce este construit tocmai peste modul în care se interacționează cu conținutul în rețelele sociale, utilizatorul evaluând explicit resursa pe care o vede, dar fără a îi fi întreruptă experiența de navigare.

6 Implementarea

În acest capitol vom detalia cum ar putea arăta o implementare a unui sistem ce deservește comunitatea online pentru care este proiectat cu metrici de încredere și reputație, dar și recomandări de conținut.

În prima secțiune vom discuta despre proprietățile pe care ar trebui să le îndeplinească un astfel de sistem, ne vom referi la scalabilitate, disponibilitate și consistență, raportându-ne atât la partea de client cât și de backend. Astfel, vom schița cerințe pentru datele necesare(cum vor fi salvate, cum le vom folosi), cum modelăm relațiile dintr-un graf al încrederii(concept pe care îl vom detalia în acest capitol) și ce servicii sunt necesare a fi furnizate din partea componentelor ce alcătuiesc acest modul pentru a fi fezabil.

În a doua secțiune vom discuta soluția propusă de autor pentru implementarea unui altfel de serviciu, vom analiza uneltele folosite în crearea soluției, modul în care au fost utilizate, dar și rezultatul final.

6.1 Proprietățile sistemului

În capitolul anterior ne-am referit în general la modul în care utilizatorul ar trebui să interacționeze cu serviciul propus, fără a intra în detalii în legătură cu proprietățile ce ar trebui îndeplinite de către componentele aplicației. Este evident faptul că va fi necesar să implementăm componente ale aplicației nu numai în cadrul backend-ului aplicației, cât și la nivel de client. În schimb va fi nevoie să transferăm date de la client către sistem pentru a calcula încrederea într-un mod cât mai precis.

6.1.1 Backend

Pentru a avea o aplicație robustă care să deservească cât mai bine clienții, trebuie să luăm în considerare cum vrem să servim informații clienților noștri.

Conform teoremei CAP[5], propusă de Brewer în anul 2000, trebuie să alegem două dintre următoarele trei proprietăți[6]:

- **„Consistență** - echivalentul existenței unei singure resurse actualizate”
- **„Disponibilitate** - a datelor pentru actualizări”
- **„Toleranță la partiționare”**

Conform teoremei, astfel, trebuie mereu să facem un compromis în ceea ce privește modul în care sistemul nostru va gestiona datele și cererile venite de la clienți. În lumea reală în schimb, în general compormisurile sunt mult mai mici decât par la prima vedere. Spre exemplu, în cazul în care se aleg disponibilitatea și toleranța la partiționare, se poate crea un model eventual consistent(adeșea întâlnit în bazele de date NoSQL).

Întrucât rețelele sociale au ajuns să aibă comunități cu dimensiuni de sute de milioane utilizatori(precum Facebook sau LinkedIn), argumentăm în acest caz importanța scalabilității unei astfel de aplicații. Din punct de vedere al costurilor, aceasta ar trebui să fie scalabilă orizontal și să asigure și elasticitate, astfel încât să nu fie nevoie ca mentenanța platformei să cauzeze lipsa furnizării de servicii clienților.

În continuare vom discuta despre posibile strategii de implementare ale unui sistem referindu-ne la necesitățile unei rețele sociale și la cele trei componente ale teoremei CAP.

În primul rând, avem nevoie de o rată mare de disponibilitate. Nu dorim ca experiența utilizatorului să aibă de suferit din cauza indisponibilității serviciului nostru. Spre exemplu, în cazul în care într-o componentă pentru fluxul de postări avem rezultate ordonate în funcție de încrederea pe care utilizatorul ar avea-o în ele, lipsa serviciului ar fi un inconvenient major pentru utilizatori.

Am menționat mai sus că avem nevoie de scalabilitate. Asta aduce odată cu ea și toleranța la partiționare. Însă, din nefericire, definițiile pentru această componentă a teoremei CAP variază, deși este singura văzută ca un predicat boolean, în timp ce disponibilitatea și consistența sunt văzute ca și valori ale unui interval.

Presupunând că alegem și toleranța la partiționare, asta ar însemna că am sacrifica consistența. Dar, acest lucru ar face sistemul nostru să returneze date ce nu sunt valide, ceea ce nu este de dorit într-un sistem pentru încredere și reputație.

Am putea să alegem o implementare eventual consistentă, acceptând o acuratețe mai mică. Pentru asta, trebuie să revedem modul în care se calculează încrederea.

$$WE_UR(U_i, R_j) = hasRead(U_i, R_j) + hasLiked(U_i, R_j) + hasCommented(U_i, R_j) + \\ + hasShared(U_i, R_j) + hasBookmarked(U_i, R_j) \quad (2)$$

$$R_{U_i} = \{r \in R | owns(r, U_i) = 1\}$$

$$WP_UU(U_i, U_j) = viewed(U_i, U_j) * 0,5 + follows(U_i, U_j) + connected(U_i, U_j) * 1,5 + endorsed(U_i, U_j) * 2$$

$$WE_UU(U_i, U_j) = \frac{\sum_{l=1}^k WE_UR(U_i, R_{U_i}^l)}{k * MaxWorth} + WP_UU(U_i, U_j) \\ 2 * MaxWorth$$

Observăm, datorită multiplilor factori implicați în calculul încrederii că o schimbare în sistem nu ar putea produce schimbări majore în rândul valorilor pentru încrederea explicită prezente în sistem. De asemenea, evenimentele pe care le generează clientul pot fi memorate la nivel de client, fiind posibile ca valorile acestea să fie actualizate la client în așteptarea ajungerii sistemului într-o stare de consistență.

Acest mod de a compensa starea de eventual consistență nu pot fi făcute și în cazul încrederii implicite. În schimb, din punct de vedere statistic, este foarte improbabil ca atât de mulți utilizatori să își schimbe rating-ul față de un utilizator într-un mod suficient de drastic încât agregarea lor să producă valori semnificativ diferite față de cele prezente. Acest lucru este cu atât mai mult valabil în calculul reputației, având în vedere că rating-urile individual au o pondere mult mai mică acolo, fiind vorba despre agregarea rezultatelor provenite de la întreaga comunitate, nu doar de la o componentă a ei.

În integrarea cu backend-ul propriu zis al rețelei sociale, trebuie să luăm în considerare și cuplajul acestei componente cu restul modulelor. Calea recomandată ar fi să avem un cuplaj cât mai redus, această componentă fiind considerată și implementată ca și un serviciu web de sine stătător, nu ca un modul în întreaga aplicație.

6.1.2 Clientul

Având în vedere complexitatea aplicației și cele menționate în secțiunea anterioară, este clar că aplicația la nivel de client joacă un rol crucial în serviciile de încredere și reputație. Acestea nu numai afișează date clientului, dar și transmite backend-ului interacțiunile clientului cu conținutul. Așadar, trebuie definite și cerințe pentru client astfel încât să putem asigura corecta furnizare a serviciului.

Astfel, clientul se face responsabil cu colectarea datelor ce folosesc mai apoi la calculul încrederii explicite. Spre deosebire de interacțiunea din perspectiva utilizatorului, acesta trebuie să monitorizeze și modul în care utilizatorul abordează conținutul afișat. Ca și consecință, trebuie urmărite mai mult evenimentele declanșate de utilizator în pagină, precum apăsarea de butoane sau navigarea printre pagini.

Pentru a urmări modelul de conversie propus, acesta trebuie să știe ce pagini sunt citite de către utilizator, fără a fi intrusiv în experiența sa, și fără a încerca să coreleze cititul cu alte evenimente(precum evenimente de tip “mouseover”).

De asemenea, acesta trebuie să înregistreze și să trimită backend-ului toate evenimentele relevante create de utilizator.

6.2 Implementarea autorului

Implementarea prezentă este un exemplu de cum poate fi aplicat modelul produs. Aceasta a fost realizată pentru a exemplifica beneficiile aduse de prezentul model într-o rețea socială profesională.

Întrucât obiectivul acestei lucrări nu a fost crearea unei rețele sociale profesionale, ci elaborarea unui model pentru încredere și reputație, implementarea a fost efectuată pentru a demonstra conceptul. Astfel ne vom limita în cele ce urmează la implementarea serviciului pentru încredere și reputație și la implementarea clientului într-un mod care să respecte cerințele secțiunii anterioare.

6.2.1 Serviciul

Serviciul furnizează date despre încredere și reputație din cadrul rețelei sociale. În implementarea prezentată, el nu doar furnizează datele, ci și stochează informațiile în forma unui graf.

Pentru asta a trebuit aleasă o bază de date implementată pe grafuri. Soluția aleasă a fost Neo4j. În continuare vom vorbi despre ea și cum se încadrează în cerințele creionate în secțiunea anterioară.

Neo4j este un sistem de management al bazelor de date pe grafuri, având stocare și procesarea datelor efectuate pe grafuri. Neo4j este scris în Java. Folosirea acestuia a ușurat modelarea rețelei încrederii prezentă în modelul StarTrust.

În cadrul Neo4j există trei tipuri de bază, acestea fiind:

- **Nod** - un nod al grafului
- **Muchie** - muchie a grafului (este orientată)
- **Atribut** - attribute ale diverselor elemente de graf, fiecare nod/muchie fiind capabil(ă) să aibă un număr nelimitat de attribute

De asemenea, trebuie menționate și etichetele, care sunt folosite pentru identifica tipuri, folosite predominant în căutări.

Prin prisma teoremei CAP[5], Neo4j, nu este tolerant la partiții. Este distribuit, în sensul că poate fi făcut un cluster de mașini care să deservească aceeași aplicație și care își distribuie încărcătura folosind un load balancer. Neo4j funcționează pe o arhitectură de tip Master/Slave.

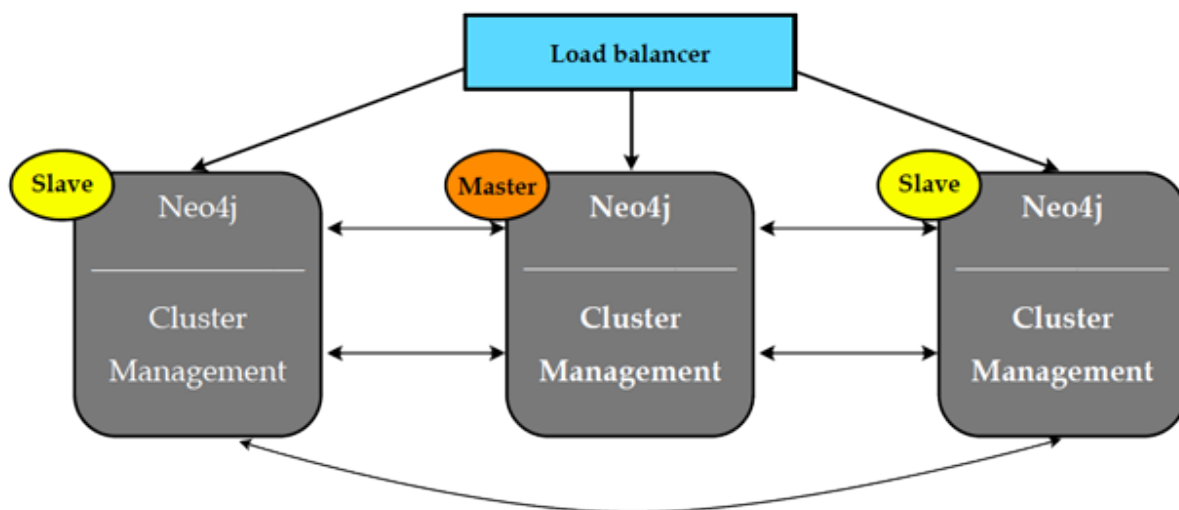


Figura 1: Arhitectura Neo4j⁵

Deficiența care îl face să nu fie tolerant la partiții este modul de stocare al datelor. Fiecare nod dintr-un cluster ce rulează Neo4j va avea în memorie întreaga bază de date. Deși este un dezavantaj major din cauza costurilor de memorie pentru o aplicație mare, acesta este un avantaj când vine vorba de performanță.

Interacțiunea cu Neo4j se face prin două metode. Una ar fi printr-un client ce implementează un protocol specific, numit bolt. Cealaltă variantă este prin intermediul unui API REST. Ambele moduri de conexiune presupun efectuarea operațiilor folosind Cypher, limbajul de interogare specific. Cypher este un limbaj declarativ de interogare, inventat de Neo Technologies pentru a fi folosit în cadrul Neo4j. Acesta a devenit Open Source în 2015.

Acesta prezintă dezvoltatorului și un mediu interactiv, accesibil prin browser, în care se pot rula căutări sau se poate verifica starea sistemului.

⁵<https://neo4j.com/neo4j-scales-web-enterprise/>

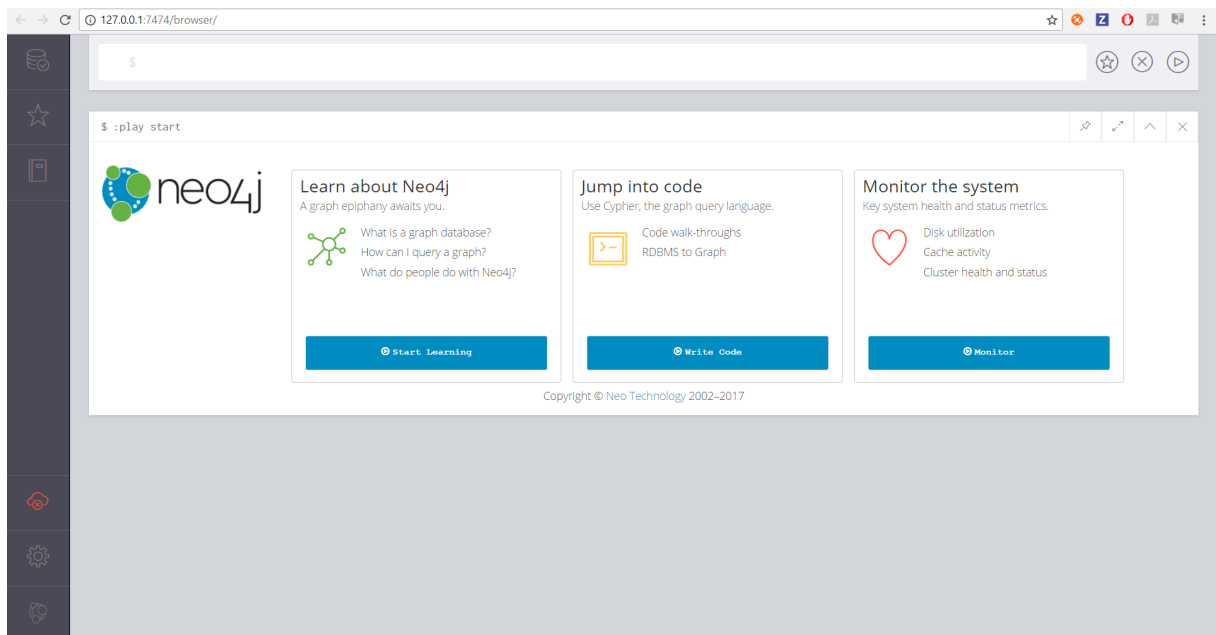


Figura 2: Mediul Interactiv Neo4j

Rezultatele căutărilor în mediul interactiv oferă multiple posibilități de a fi vizualizate. Astfel, rezultatele pot fi văzute tabular, în format JSON (cum sunt stocate, Neo4j fiind o bază de date NoSQL), sau grafic, fiind reprezentat un subgraf conținând doar nodurile și muchiile căutate.

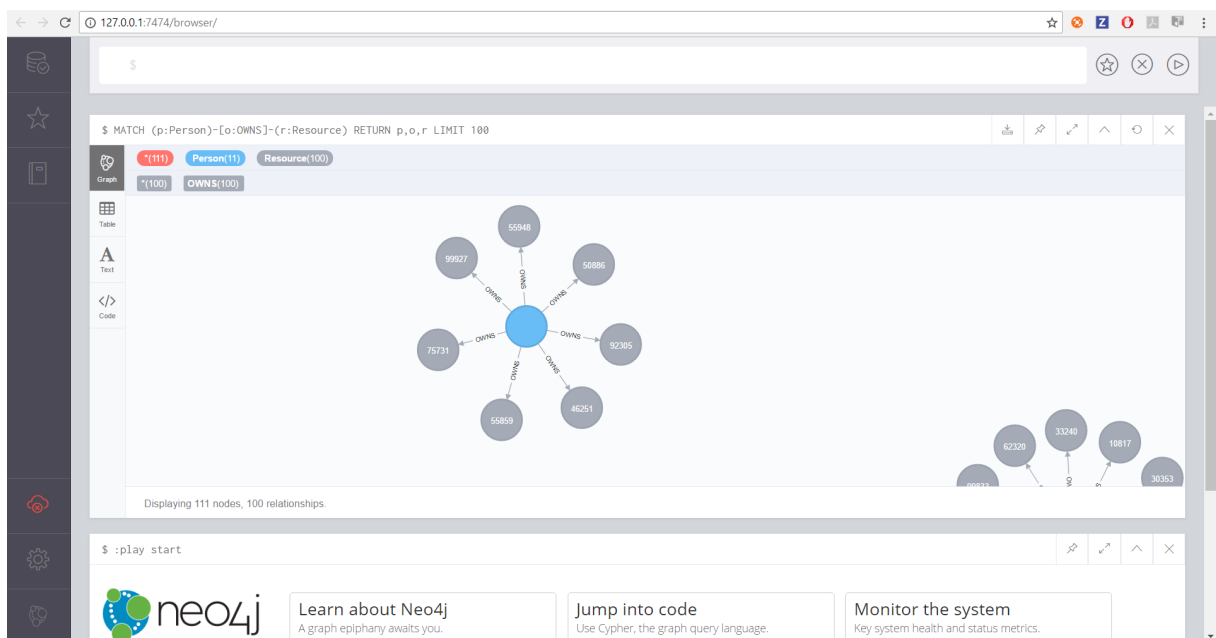


Figura 3: Vizualizarea rezultatelor

6.2.2 Clientul

Clientul este o aplicație ce simulează o rețea socială profesională. Datele sunt reprezentate similar modelului din backend. Am folosit pentru implementarea aplicației framework-ul An-

gular, fiecărui tip de date fiindu-i asociat o componentă individuală. Am folosit o librărie ce monitorizează comportamentul utilizatorilor, urmărind elementele care sunt afișate utilizatorilor.

Framework-ul Angular

Angular este un framework MVW* ce facilitează dezvoltarea aplicațiilor clienți. Acesta oferă șabloane declarative, injectarea dependențelor, utilitate end-to-end și soluții integrate pentru a ajuta dezvoltatorul în rezolvarea problemelor din ciclul de viață al unei aplicații. Acesta este disponibil atât pentru limbajul de scripting TypeScript, cât și pentru JavaScript. În cazul soluției pentru TypeScript, programarea este bazată pe un meta-limbaj oferit de către aceștia.

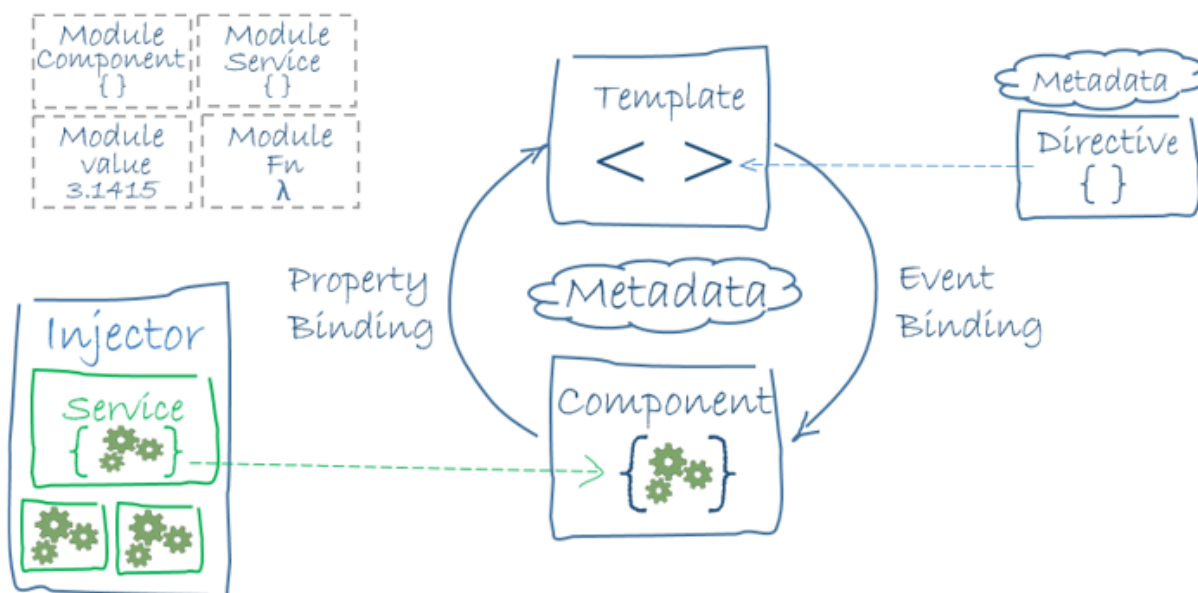


Figura 4: Arhitectura Angular⁶

Acesta propune un model format din mai multe clase de resurse. Dintre acestea, le menționăm pe cele mai importante:

- **Module** - Fiecare aplicație are minim un modul, acesta fiind numit modulul rădăcină(engl. *rootModule*). După cum vom vedea din exemplele de cod ce urmează, un modul este responsabil în general cu gestionarea conținutului său, ce de obicei desemnează o funcționalitate sau un singur scenariu.
- **Componente** - O componentă controlează o bucată din ecran numită *view*. Framework-ul crează, actualizează și distruge componente în funcție de interacțiunea utilizatorului cu

⁶<https://angular.io/guide/architecture>

aplicația. Aceste moduri de control al componentelor poate fi adaptat nevoilor sale prin hook-uri ce controlează ciclul de viață al componentei

- **Template-uri** - Modul de vizualizare al unei componente este definit cu ajutorul template-ului asociat. Template-ul este o formă de HTML ce este îmbunătățită folosind metalimbajul caracteristic al framework-ului, permițând dezvoltatorului să personalizeze modul de afișare în funcție de diferite scenarii.
- **Servicii** - Serviciile sunt clase ce oferă diferite funcționalități. Acestea pot fi de orice natură, de la împachetarea unui client HTTP pentru un API folosit la o clasă ce face operații matematice.

În afară de framework-ul Angular, am folosit și librăria TimeMe.js⁷. Aceasta urmărește cât de mult utilizatorul interacționează cu o pagină sau cu componentele acesteia. Pentru a monitoriza interacțiunea se instanțiază un contor pentru ea folosind apelarea unei metode din librărie. Cu ajutorul ei am implementat modelul de conversie al utilizatorului, folosind timpul de vizualizare al unei componente ce desemnează o resursă pentru a monitoriza partea de **Interaction** din fluxul de conversie.

Pentru a comunica cu serviciul s-a apelat la clientul JavaScript pentru Neo4j. Acesta a fost integrat cu TypeScript-ul (TypeScript este un superset al JavaScript, ce adaugă suport pentru clase și multe altele). Acesta a fost împachetat într-un serviciu Angular, ce era oferit la rădăcina aplicației (în componenta aplicației propriu-zise). Am făcut această alegere deoarece la instanțierea unui serviciu într-o componentă se parcurge arborele aplicației în sens invers până la rădăcină sau până ce este întâlnit serviciul. Faptul că am ales ca serviciul să fie oferit de către nodul rădăcină duce la tratarea sa ca un Singleton, fiecărei componente a aplicației fiindu-i oferit un pointer către instanța ce se află în componenta de bază.

```
import {Component, Input, OnDestroy, OnInit} from '@angular/core';
import {Post} from "../../model/post.model";
import {Neo4jService} from "../../neo4j.service";
import {User} from "../../model/user.model";

import * as TimeMe from 'timeme.js';

@Component({
```

⁷<https://github.com/jasonzissman/TimeMe.js/tree/master>

```

    selector: 'postcard',
    templateUrl: './post.component.html',
    styleUrls: ['./post.component.css']
  })
  export class PostComponent implements OnInit, OnDestroy {
    private bookmarked: boolean;
    private neo4jService: Neo4jService;

    @Input() post: Post;
    @Input() user: User;

    constructor(neo4jService: Neo4jService) {}

    ngOnInit(): void {
      TimeMe.trackTimeOnElement(this.post.id.toString());
    }

    ngOnDestroy(): void {
      let liked = this.post.likedBy.find(user => user === this.user) == null;
      let comment = this.post.comments.find(comment => comment.user === this.user);
      let shared = this.post.sharedBy.find(user => user === this.user) == null;
      let viewed = liked || comment != null || shared || this.bookmarked ||
        TimeMe.getTimeOnElementInSeconds() >= 10;

      this.neo4jService.addPostEvaluation(
        this.user.id,
        this.post.id,
        viewed, liked,
        comment == null ? null : comment.content,
        shared,
        this.bookmarked
      );
    }
  }
}

```

În codul de mai sus aprătam cum se definește o componentă a unei aplicații Angular. Se observă adnotarea `@Component` împreună cu caracteristicile sale definite într-un obiect atașat adnotării. De asemenea, am ilustrat și folosirea librării `TimeMe.js` pentru a monitoriza interacțiunile utilizatorului cu conținutul postării. Apariția a `neo4jService` în constructorul clasei arată procedura de injectare a unui serviciu în cadrul unei componente. Se pot observa și metodele **`ngOnInit`** și **`ngOnDestroy`**. Acestea fac parte din hook-urile de pe durata vieții ce pot fi modificate de către utilizator, **`ngOnInit`** fiind momentul randării componente, în care apelăm metoda librării `TimeMe.js` **`trackTimeOnElement()`** pentru a începe să urmărim interacțiunea utilizatorului cu actualizarea reprezentată cu ajutorul prezentei clase. Folosind **`ngOnDestroy`** salvăm interacțiunile utilizatorului cu post-ul prin intermediul serviciului ce interacționează cu componenta de backend.

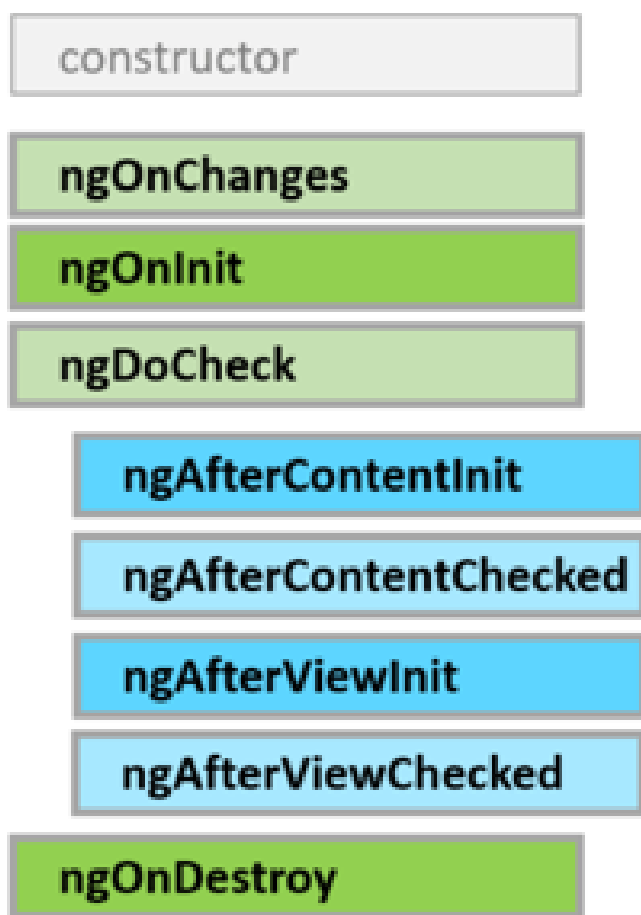


Figura 5: Hook-uri Angular pe durata vieții unei componente⁸

```
<div class="row" style="margin-top: 30px">
  <div class="col-3">
    <profile-widget [user]='getUser()'></profile-widget>
```

⁸<https://angular.io/guide/lifecycle-hooks>

```

</div>
<div class="col">
    <div *ngFor="let post of getPosts()" style="margin-top: 10px">
        <postcard [post]="post" [user]="getUser()" class="mt-3">
        </postcard>
    </div>
</div>
<div class="col-3">
    <recommendation-widget [user]='getUser()'>
    </recommendation-widget>
</div>
</div>

```

Pentru stilizarea conținutului am folosit librăria Bootstrap. În exemplul de mai sus se observă limbajul de markup specific aplicațiilor Angular. Astfel, observăm că fiecare componentă are propriul ei selector. Demn de menționat este și modul ușor de a instanția multiple componente de același tip folosind directiva **ngFor**. În exemplul de mai sus se observă și modul de transmisie a datelor. Urmărind și codul din clasa PostComponent, putem observa că folosind adnotarea **@Input** putem inițializa componente cu valori transmise de componentele părinte.

```

import { NgModule } from '@angular/core';
import { CommonModule } from '@angular/common';
import { PostComponent } from '../post/post.component';
import { FeedComponent } from '../page/page.component';
import { ProfileModule } from '../profile/profile.module';

@NgModule({
    imports: [
        CommonModule,
        ProfileModule
    ],
    declarations: [PostComponent, FeedComponent]
})
export class FeedModule { }

```

În exemplul de mai sus avem declarația unui modul ce conține doar declarații de componente, respectiv expresii de import ale altor module.

```
@NgModule({
  imports: [
    CommonModule
  ],
  declarations: [ProfilePageComponent, ProfileWidgetComponent],
  exports: [ProfileWidgetComponent]
})

export class ProfileModule { }
```

În cadrul acestui modul putem observa și apariția expresiei **exports**. Aceasta este utilizată pentru a permite altor module să importe anumite componente din modul, permițând astfel dezvoltatorului să creeze module ce au rol de librării.

Concluzii

Oamenii sunt în prezent mai conectați la Internet decât niciodată. Este foarte posibil ca într-un viitor nu foarte distant circa două miliarde de oameni să fie înscrși pe rețele de socializare. Astfel, companiile precum Facebook ajung să fie depozite uriașe de informație, depășind sute de petabytes ce reprezintă conținut generat de oameni. În schimb, pentru a menține utilizatorii cât mai activi pe astfel de platforme, rețelele sociale trebuie să le arate utilizatorilor conținutul cât mai relevant pentru el. Astfel, a apărut această nevoie de optimizare a modurului în care conținutul este oferit spre consum.

Există enorm de multe posibilități în algoritmi de recomandare, de încredere, de reputație, algoritmi ce pot ordona conținut după relevanța pentru utilizator, într-un mod personalizat. Iar indiferent cu ce rată va crește cantitatea medie de informație produsă de un individ, nevoia pentru algoritmi de acest tip va crește direct proporțional cu ea. Dar având în vedere curente precum Internet of Things și Wearables, putem doar presupune că odată cu apariția și, mai important, adopția lor, informațiile vor cunoaște noi dimensiuni, mai mari decât niciodată.

Dar, odată cu aceste trenduri vor apărea și noi oportunități pentru algoritmii de acest tip, nu numai pentru recomandarea de conținut digital. Cu cât suntem mai conectați la senzori, cu atât avem mai multe date de oferit, dar și de primit.

Din această cauză consider că un astfel de algoritm poate fi adaptat pentru a ajuta în domeniul sanitar. Folosind un sistem bazat pe un graf al încrederii, o persoană alergică și-ar putea alergiile astfel încât să știe exact ce ar putea să îi facă rău sau nu. Medicii ar putea oferi tratamente pe baza sistemelor de acest tip, fiind modelate răspunsurile la alte tratamente ale unei persoane și relațiile dintre ele.

Într-o eră în care securitatea în mediul digital este critică tocmai din cauză că portabilitatea dispozitivelor a făcut ca aproape toate datele noastre, fie ele de la senzori precum cele ce măsoară frecvența cardiacă sau modul în care dormim, sau senzori foto ce sunt acum în uzul fiecărei persoane ce deține un terminal mobile, securitatea datelor noastre este mereu o problemă. Astfel, un sistem de acest tip ar putea modela o rețea a încrederii, care, pe bază de algoritmi precum cei de searchable encryption, să securizeze datele într-un mod în care utilizatorul să aibă control deplin asupra datelor.

Prin modelul propus de prezenta lucrare am dorit să demonstrăm că putem avea un algoritm de încredere și reputație care să ofere utilizatorilor un conținut optimizat pentru ei, dar care să nu ia din timpul redus de căutare a resurselor și utilizatorilor cu semnificație mai mare pentru client și să refolosească acel timp pentru a-l obliga să ofere rating-uri celorlalți utilizatori pentru

a obține această experiență.

Acest model a fost proiectat întocmai pentru a avea un sistem ce evoluează în mod natural odată cu utilizatorii, fără a le perturba experiența de navigare. De asemenea, întrucât platformele sociale sunt construite pe baza conținutului făcut public de către utilizatori, am decis să îi oferim o proporție mai mare în modelul nostru, întrucât considerăm că acest tipuri de aplicații sunt extrem de axate pe conținut.

Bibliografie

- [1] D. M. Boyd and N. B. Ellison, “Social network sites: Definition, history, and scholarship,” *Journal of Computer-Mediated Communication*, no. 13, pp. 210–230, 2008.
- [2] J. Golbeck, “Computing and applying trust in web-based social networks phd thesis, university of maryland,”
- [3] L. Alboaie and M.-F. Vaida, “Trust and reputation model for various online communities,”
- [4] T. Berners-Lee, R. Fielding, and L. Masinter, “Uniform resource identifiers (uri): Generic syntax,” August 1998.
- [5] E. Brewer, “Towards robust distributed system,” *Symposium on Principles of Distributed Computing (PODC)*, 2000.
- [6] E. Brewer, “Cap twelve years later: How the ”rules” have changed,” <https://www.infoq.com/articles/cap-twelve-years-later-how-the-rules-have-changed>, 2012.
- [7] “<https://code.facebook.com/posts/229861827208629/scaling-the-facebook-data-warehouse-to-300-pb/>.”
- [8] “<https://www.youtube.com/watch?v=7smctfyxw8u>.”
- [9] “<http://www.telegraph.co.uk/technology/2017/05/03/facebook-approaches-2-billion-users/>.”
- [10] “<https://blog.linkedin.com/2017/april/24/the-power-of-linkedin-500-million-community>.”
- [11] “<https://neo4j.com/neo4j-scales-web-enterprise/>.”
- [12] “<https://angular.io/guide/architecture>.”
- [13] “<https://github.com/jasonzissman/timeme.js/tree/master>.”
- [14] “<https://v4-alpha.getbootstrap.com>.”