



**UNIVERSITATEA NAȚIONALĂ  
DE ȘTIINȚE ȘI TEHNOLOGIE  
POLITEHNICA BUCUREȘTI**



**Școala Doctorală de Electronică, Telecomunicații și  
Tehnologia Informației**

**Decizie nr. 94 din 05-10-2023**

**REZUMAT TEZĂ  
DE DOCTORAT**

**Ing. Adrian PETCU**

---

**MIGRAREA CĂTRE WEB3.0: IMPACTUL SCALĂRII ȘI  
DESCENTRALIZĂRII ASUPRA APLICAȚIILOR SOFTWARE**

**MIGRATING TO WEB3.0: THE IMPACT OF SCALING AND  
DECENTRALISATION ON SOFTWARE APPLICATIONS**

---

**COMISIA DE DOCTORAT**

<b>Prof. Dr. Ing. Mihai Alexandru CIUC</b> Univ. Politehnica din București	Președinte
<b>Prof. Dr. Ing. Dan Alexandru STOICHESCU</b> Univ. Politehnica din București	Conducător de doctorat
<b>Prof. Dr. Ing. Dan Marius DOBREA</b> Univ. Tehnică din Iași	Referent
<b>Prof. Dr. Ing. Cristian GRAVA</b> Univ. din Oradea	Referent
<b>Conf. Dr. Ing. Bogdan Cristian FLOREA</b> Univ. Politehnica din București	Referent

**BUCUREȘTI 2023**

---

# Cuprins

<b>1</b>	<b>Introducere</b>	<b>1</b>
1.1	Prezentarea domeniului tezei de doctorat . . . . .	1
1.1.1	Web3.0 și descentralizare . . . . .	2
1.2	Motivație . . . . .	2
1.3	Scopul tezei de doctorat . . . . .	3
1.4	Conținutul tezei de doctorat . . . . .	3
<b>2</b>	<b>Tendințe actuale și lucrări conexe</b>	<b>5</b>
2.1	Evoluția aplicațiilor . . . . .	5
2.1.1	Limbaje de programare și lucru colaborativ . . . . .	5
2.1.2	Distribuirea aplicațiilor . . . . .	6
2.1.3	Scalarea interfețelor grafice web . . . . .	6
2.2	Blockchain și descentralizare . . . . .	6
2.2.1	Descentralizare pre-blockchain . . . . .	6
2.2.2	Descentralizare în era blockchain . . . . .	6
2.2.3	Economia descentralizată . . . . .	7
2.2.4	Web3.0 . . . . .	7
<b>3</b>	<b>Fundamente teoretice</b>	<b>8</b>
3.1	Aplicații, organizare și distribuie . . . . .	8
3.1.1	Conținutul pachetului de aplicații . . . . .	8
3.1.2	Tipuri de aplicații . . . . .	8
3.2	Tipuri de arhitectură a aplicațiilor . . . . .	9
3.2.1	Arhitectura Monolitică . . . . .	9
3.2.2	Arhitectura bazată pe microservicii . . . . .	9
3.2.3	Monoliți și microservicii; analiză comparativă . . . . .	9
3.3	Descentralizare folosind Blockchain . . . . .	9
3.3.1	Blockchain-uri private . . . . .	10
3.3.2	Blockchain-uri publice . . . . .	10
3.3.3	Stocare descentralizată (IPFS) . . . . .	10
3.3.4	Contracte Inteligente . . . . .	10
3.3.5	Mecanisme de consens . . . . .	10

3.3.6	Securitate și vectori de atac . . . . .	10
3.3.7	Împuternicirea utilizatorilor și guvernanta datelor . . . . .	11
3.3.8	Modele de cost . . . . .	11
3.4	Portofele electronice . . . . .	11
3.4.1	Tipuri de portofele electronice . . . . .	11
3.4.2	Securitatea portofelelor electronice . . . . .	11
3.4.3	Interacțiunea cu portofelele electronice . . . . .	11
3.5	Web 3.0 . . . . .	12
3.5.1	Arhitectura Web 3.0 . . . . .	12
3.5.2	Aplicații Web 3.0 . . . . .	12
<b>4</b>	<b>Migrarea de la aplicații instalabile nativ la aplicații web</b>	<b>13</b>
4.1	Experiența utilizatorului și provocările tehnice . . . . .	13
4.1.1	Experiența utilizatorului . . . . .	13
4.1.2	Cerințe tehnice . . . . .	13
4.1.3	Avantajele și dezavantajele aplicațiilor instalate de utilizator . . . . .	14
4.2	Aspecte cheie în tranziția către o infrastructură descentralizată . . . . .	14
4.3	Analiză comparativă a performanței unei aplicații web disponibilă și în format instalabil . . . . .	14
4.4	Concluzii . . . . .	16
<b>5</b>	<b>Scalarea interfețelor grafice ale aplicațiilor web folosind micro-frontend</b>	<b>18</b>
5.1	Micro-frontend . . . . .	18
5.1.1	Tipuri de compoziție . . . . .	18
5.1.2	Provocările utilizării unei arhitecturi micro-frontend . . . . .	18
5.1.3	Beneficii . . . . .	19
5.1.4	Soluții Micro-Frontend . . . . .	19
5.2	Analiza performanței diverselor tipuri de soluții micro-frontend . . . . .	19
5.3	Concluzii . . . . .	21
<b>6</b>	<b>Web3.0 și descentralizare; aplicații practice</b>	<b>22</b>
6.1	Autentificare descentralizată folosind web 3.0 . . . . .	22
6.1.1	Mecanisme de Autentificare Securizate . . . . .	22
6.1.2	Componentele sistemului . . . . .	22
6.1.3	Autentificarea în era Web 2.0 . . . . .	23
6.1.4	Autentificarea în era Web 3.0 . . . . .	23
6.1.5	Implementarea sistemului de autentificare folosind Web3.0 . . . . .	23
6.1.6	Analiza comparativă de performanță a metodelor de autentificare . . . . .	23
6.1.7	Concluzii . . . . .	25
6.2	Implementarea practică a unui sistem de semnare digitală a documentelor folosind tehnologia blockchain . . . . .	25

6.2.1	Platforme pentru semnături digitale . . . . .	25
6.2.2	Tipuri de semnare a documentelor . . . . .	27
6.2.3	Semnături digitale . . . . .	27
6.2.4	Hash-uri și sume de control . . . . .	27
6.2.5	Implementarea sistemului propus . . . . .	28
6.2.6	Concluzii . . . . .	31
<b>7</b>	<b>Concluzii</b>	<b>32</b>
7.1	Obiective generale și rezultate obținute . . . . .	32
7.2	Contribuții originale . . . . .	34
7.3	Lista lucrărilor originale . . . . .	35
7.4	Perspectivă de dezvoltare ulterioară . . . . .	36
	<b>Bibliografie</b>	<b>38</b>

# Capitolul 1

## Introducere

Pe măsură ce tehnologia se integrează tot mai mult în viața noastră de zi cu zi, ea transformă în mod fundamental percepția, utilizarea și dependența noastră față de aplicații și platforme digitale. Modul în care folosim aplicațiile software a evoluat semnificativ în timp prin punerea accentului pe accesibilitate și scalabilitate.

În cadrul acestui proces dinamic, a avut loc o trecere de la platformele convenționale care necesită instalare locală la interfețe bazate pe web. Această tranziție a adus o nouă eră caracterizată de o accesibilitate sporită și o disponibilitate pe scară largă. Procesul de tranziție nu se limitează doar la schimbări de platformă, ci include mai degrabă o reevaluare mai cuprinzătoare a funcționalităților software și a principiilor de proiectare. Mai mult, apariția micro-front-end-urilor a dus la o îndepărtare de proiectele monolitice, acordând prioritate structurilor modulare și scalabile care răspund cerințelor variate și în continuă evoluție ale consumatorilor moderni. Această tranziție a dus la *centralizarea* informațiilor utilizatorilor și nevoia de *încredere* în reziliența și securitatea platformelor web, spre deosebire de stocarea datelor local.

Scopul prezentului studiu este investigarea schimbărilor, cu un accent specific pe migrarea, scalabilitatea și *descentralizarea* sistemelor software.

### 1.1 Prezentarea domeniului tezei de doctorat

Apariția internetului a revoluționat modul în care accesăm și partajăm informații și a deschis lumea unui univers prin intermediul căruia putem interacționa și influența proiecte, produse și legături sociale. Companiile private au realizat că modul de distribuție și de control al software-ului (sub o licență) poate fi realizat cu ușurință prin expunerea de interfețe grafice sub forma unor pagini web, spre deosebire de oferirea de pachete instalabile utilizatorilor. Totuși, această schimbare de direcție, a venit cu provocări tehnice și limitări. La începutul creșterii în popularitate a internetului, acesta era folosit de către companiile private pentru distribuirea de pachete instalabile utilizatorilor săi, rezolvând astfel problema distribuției software-ului și renunțând la suportul hardware.

Următorul pas în evoluția aplicațiilor software a dus la reorientarea strategică către platforme web și înlocuirea pachetelor instalabile pe dispozitivele utilizatorilor.

Crearea de aplicații web care imită în totalitate comportamentul celor instalabile s-a dovedit a fi o provocare întrucât browserele web erau slab performante și aveau multe limitări.

Deși sunt multe beneficii în migrarea aplicațiilor către domeniul web, pot exista și dezavantaje. Totuși, dacă performanța aplicației web este comparabilă sau chiar mai bună decât aceea a aplicației instalate local, putem să considerăm că beneficiile sunt mult mai mari decât dezavantajele. Un prim subiect de cercetare este analiza performanței unei aplicații disponibile în ambele variante, atât în varianta instalabilă pe dispozitivul hardware, cât și în versiunea online.

Tehnologia cloud a revoluționat dezvoltarea aplicațiilor online, permițând organizațiilor să stocheze și să analizeze date fără să gestioneze servere fizice, facilitând scalabilitatea și adaptabilitatea. Cu toate acestea, trecerea la arhitecturi bazate pe microservicii a evidențiat un punct nevralgic: frontend-ul monolitic. Acesta limita flexibilitatea și viteza de dezvoltare, însă introducerea conceptului de micro-frontend a abordat această problemă, permițând dezvoltarea componentelor GUI în mod autonom și scalabil, fără interferențe.

### **1.1.1 Web3.0 și descentralizare**

Privind în viitor, conceptul de Web3.0 și descentralizarea reprezintă un nou orizont în dezvoltarea aplicațiilor web. Web3.0 pune accent pe descentralizare, securitate, transparență și controlul utilizatorului asupra datelor proprii. Utilizarea tehnologiilor precum blockchain și contracte inteligente va permite crearea unor aplicații web mai sigure și transparente. Cu toate acestea, realizarea acestei viziuni necesită o colaborare continuă, cercetare și dezvoltare [1] pentru a aborda provocările tehnice, etice și sociale care pot apărea pe parcurs. Potențialul Web3.0 de a remodela internetul într-un mod care servește mai bine interesul public și individual face ca această explorare să fie nu doar valoroasă, ci și esențială pentru viitorul digital al umanității.

## **1.2 Motivație**

Evoluția dezvoltării software-ului a fost caracterizată de un proces perpetuu de invenție și ajustare. Peisajul dezvoltării software-ului a cunoscut transformări semnificative, pornind de la epoca aplicațiilor instalate nativ, trecând la aplicațiile bazate pe web și avansând acum spre frontiera aplicațiilor descentralizate care utilizează tehnologia Web3.0. Revoluția menționată mai sus nu a modificat doar modul în care este dezvoltat software-ul, ci și modul în care îl înțelegem și ne implicăm în el.

Această teză își propune să ofere o imagine de ansamblu a aspectelor istorice, actuale și viitoare ale dezvoltării de software, examinând trecerea de la aplicațiile instalate nativ la aplicațiile web, analizând rolul micro-frontend-urilor în abordarea problemelor de scalabilitate și explorând potențialul Web3.0 și al descentralizării.

### 1.3 Scopul tezei de doctorat

Luând în considerare aspectul evolutiv al aplicațiilor software, putem considera analiza istorică a evoluției acestora odată cu trecerea timpului ajungând la o radiografie a timpurilor curente și o analiză a tendințelor viitoare prin prisma descentralizării.

Putem considera deci că în următoarele capitole ne dorim să îndeplinim următoarele țeluri:

- realizarea unui studiu cu privire la evoluția aplicațiilor din mediul nativ în mediul online;
- identificarea problemelor potențialelor și găsirea de soluții pentru acestea;
- cercetarea și implementarea unor soluții populare de scalare a interfețelor grafice utilizator;
- cercetarea implicațiilor descentralizării ca pas evolutiv al aplicațiilor software. Aplicații practice și implementări practice;
- obținerea de concluzii referitoare la subiectele cercetate;

### 1.4 Conținutul tezei de doctorat

Teza începe cu un capitol introductiv în care sunt prezentate domeniul cercetat, motivația și scopul studiului curent.

**Capitolul 2** prezintă tendințe curente și lucrări conexe; subiectele studiate sunt de ultimă generație și cercetarea acestora în lumea academică este în plină creștere. Soluții alternative și implementări simple ale acestora sunt prezentate și comparate.

**Capitolul 3** descrie fundamentele teoretice ale aplicațiilor web și diversele modalități de împachetare a acestora dar și sisteme de arhitectura software-ului populare. Nu în ultimul rând, sunt prezentate concepte fundamentale ce stau la baza sistemelor de tip blockchain și aplicațiilor descentralizate în lumea Web3.0. Toate aceste aspecte teoretice sunt utilizate ca bază a cercetării desfășurate în această teză.

**Capitolul 4** introduce conceptul migrării aplicațiilor instalabile nativ către aplicații web împreună cu provocările și avantajele aduse de acest proces. Elementul central al acestei secțiuni este analiza comparativă de performanță dintre două variante ale aceeași aplicații (Web și instalabil nativ) folosind criterii bine stabilite. Capitolul continuă cu

validarea metodologiei de cercetare și cu realizarea experimentului propus. În final, rezultatele obținute sunt analizate și, pe baza lor, sunt prezentate recomandări și concluzii.

**Capitolul 5** analizează în detaliu diferitele soluții populare de micro-frontend-uri, evaluând avantajele și dezavantajele fiecăreia, și oferă orientări practice pentru implementarea acestora în cadrul organizațiilor moderne care își doresc să îmbunătățească experiența utilizatorilor în mediul online. Experimentul practic constă în implementarea a două soluții populare și compararea performanței acestora raportată la implementarea de bază (monolit). Rezultatele obținute sunt centralizate și analizate iar pe baza lor sunt redactate concluzii, recomandări și perspective de viitor.

**Capitolul 6** oferă o analiză profundă a conceptelor, tehnologiilor și arhitecturilor care stau la baza *Web3.0*, inclusiv contracte inteligente, cripto monede și rețele peer-to-peer. De asemenea, sunt prezentate studii de caz și exemple concrete care ilustrează cum pot fi utilizate aceste tehnologii pentru a crea aplicații mai sigure, transparente și rezistente, care să ofere mai mult control atât persoanelor individuale, cât și comunităților.

În final, **Capitolul 7** prezintă concluziile finale, însoțite de o prezentare succintă a contribuțiilor prezentate în teză și a unor căi de cercetare viitoare posibile.



# Capitolul 2

## Tendențe actuale și lucrări conexe

### 2.1 Evoluția aplicațiilor

De-a lungul timpului, limbajele de programare au suferit modificări fundamentale în ceea ce privește posibilitățile lor, ușurința de a dezvolta noi programe[2] și modul colaborativ de lucru. O dată cu creșterea popularității site-urilor web, în anii 2000, limbaje de programare dinamice cum ar fi Ruby și Javascript au luat avânt datorită concentrării lor pe ușurința dezvoltării în spațiul web.

Începând cu anul 2010, și-au făcut apariția numeroase limbaje de programare. În domeniul descentralizării aplicațiilor, cele mai notabile sunt Rust și Solidity. Limbajul de programare Rust dezvoltat de Mozilla în 2010, a fost conceput ca limbaj de programare pentru sisteme sigure și performante [3]. Solidity, creat de consorțiul Ethereum în 2014, servește ca limbaj orientat pe contracte pentru scrierea de contracte inteligente pe blockchain-ul Ethereum. În timp ce Rust excelează în construirea sistemelor eficiente și sigure, Solidity joacă un rol crucial în guvernarea comportamentului aplicațiilor descentralizate prin smart contracte auto-executabile..

#### 2.1.1 Limbaje de programare și lucru colaborativ

Pe lângă evoluția în timp a popularității limbajelor de programare, trebuie să menționăm că și lucrul colaborativ a suferit modificări pozitive. Apariția sistemelor de control al versiunii codului au revoluționat modul în care dezvoltarea software poate fi distribuită între mai mulți participanți la soluție. Git este folosit în echipe de dezvoltare mari, în care membrii echipei acționează oarecum independent și sunt răspândiți pe o arie geografică mare [4].

Alegerea limbajului de programare preferat a fost impusă de nevoia ușurinței de a scrie cod, productivității crescute și securității sporite. Lucrul colaborativ și programele cu sursă deschisă au contribuit la evoluția limbajelor [5] de programare transformând modul în care scriem codul și deschiderea drumului către dezvoltarea de soluții complexe, inovative și scalabile.

## **2.1.2 Distribuirea aplicațiilor**

Evoluția distribuției soluțiilor software a văzut o schimbare de paradigmă de la procurarea dispozitivului fizic pe care aceasta era stocată, la descărcarea soluției de pe internet iar în zilele curente, la accesarea soluției software sub forma de aplicație web.

## **2.1.3 Scalarea interfețelor grafice web**

Crearea de aplicații web cu o bază de utilizatori diversificată este o componentă esențială a dezvoltării web contemporane. Prin urmare, aplicațiile web trebuie să fie create de la zero pentru a fi scalabile, pentru a garanta că pot gestiona un eventual aflax crescut de utilizatori.

Aplicațiile grafice utilizator (GUI) au tendința să devină foarte greu de întreținut întrucât numeroase echipe pot dezvolta în același timp funcționalități noi ce pot atinge aceeași bucată de cod. În capitolul 5 explorăm diferite tipuri de scalare a interfețelor grafice utilizator.

## **2.2 Blockchain și descentralizare**

Apariția tehnologiei blockchain a modificat substanțial diferite sectoare, de la cel bancar la cel al sănătății. O metodă eficientă, descentralizată de gestionare a datelor și de procesare a tranzacțiilor, securizată criptografic prin natură, transparentă și imutabilă, tehnologia blockchain promite un răspuns perturbator la multe dintre dificultățile actuale.

Blockchain, în esența sa, este o tehnologie de registru distribuit (DLT) care asigură că înregistrările sunt atât imutabile, cât și transparente. Spre deosebire de bazele de date centralizate tradiționale, în care controlul este deținut de o singură entitate, într-un sistem descentralizat, controlul este difuzat între participanții săi.

### **2.2.1 Descentralizare pre-blockchain**

Sistemele peer-to-peer (P2P) au reprezentat o formă de descentralizare înainte ca tehnologia blockchain să fie conceptualizată și utilizată pe scară largă.

Stabilirea încrederii în sistemele P2P este crucială și, spre deosebire de sistemele centralizate unde încrederea este legată de o entitate centrală, P2P necesită un model de securitate dinamic și descentralizat. Un model de securitate descentralizat este prezentat de către autori în [6].

### **2.2.2 Descentralizare în era blockchain**

Blockchain vizează descentralizarea încrederii și consensului, oferind registre imutabile și transparente. Această tehnologie nu doar promite o securitate mai mare, ci reprezintă

o posibilă amenințare la adresa sistemelor energetice tradiționale centralizate. Introduce potențialul pentru descentralizarea schimburilor financiare, contractelor și chiar guvernului, susținând tranzacții care se bazează pe dovezi criptografice, fără intermediari. Instituții de prim rang, precum Statele Unite și Uniunea Europeană, explorează descentralizarea monetară, cu inițiative precum Digital Euro [7] și Digital Dollar [8], deși adopția integrală necesită considerare atentă a riscurilor. În afara sectorului economic, blockchain poate aduce beneficii în sfera publică și privată, de la identități digitale și votare securizată la avantaje pentru companii, deși există preocupări legate de securitate.

### **2.2.3 Economia descentralizată**

Criptomonedele au deschis calea către un nou sistem de transfer și schimb digital, având potențialul de a digitaliza și descentraliza economiile. Implementarea unor monede digitale precum dolarul și euro, bazate pe tehnologia blockchain, poate revoluționa peisajul economic, sporind eficiența, accesibilitatea și transparența financiară. Cu toate acestea, în această tranziție, este esențial să se abordeze cu prudență aspectele de confidențialitate, securitate și control.

### **2.2.4 Web3.0**

Web3.0, sau web-ul descentralizat, a câștigat teren de curând, cu accent pe descentralizare și oferirea unui control mai mare utilizatorilor asupra datelor și identității lor. Această viziune contrastează cu modelul Web2.0, dominat de câteva companii tech mari care controlează serviciile și datele. Cu ajutorul blockchain-ului, Web3.0 se îndreaptă spre un sistem unde încrederea este descentralizată și utilizatorii au mai mult control. Există discuții despre cum Web3.0 și Web2.0 pot coexista și colabora, dar pot fi, de asemenea, în competiție directă

# Capitolul 3

## Fundamente teoretice

### 3.1 Aplicații, organizare și distribuire

O aplicație este un material (software sau hardware) conceput pentru a satisface nevoi specifice sau cu rol de divertisment pentru utilizatorii finali. Inițial, termenul "aplicație" în lumea software se referea la pachete de cod scrise în diverse limbaje de programare (C++, Java, etc.) destinate să fie executate pe un dispozitiv de către un sistem de operare (Windows / Linux). Ulterior, cu creșterea în popularitate a internetului, aplicațiile au migrat către interfețe web.

#### 3.1.1 Conținutul pachetului de aplicații

Aplicațiile sunt, în general, compuse dintr-un fișier executabil, biblioteci și resurse statice. În funcție de tehnologie, aceasta poate fi reprezentată fie printr-un fișier executabil specific (cum ar fi .exe, .sh, .dmg), fie printr-un fișier HTML. Procesul de instalare a unei aplicații presupune dezarhivarea unui fișier într-un anumit loc și configurarea intrărilor necesare în sistemul de operare pentru funcționarea sa corespunzătoare [9].

#### 3.1.2 Tipuri de aplicații

Aplicațiile se bazează pe sisteme de operare pentru a furniza funcționalitatea de bază și accesul la hardware. Odată cu expansiunea rapidă a internetului și a beneficiilor sale, mai multe tipuri de aplicații au devenit disponibile pentru utilizatorii finali întrucât modul de distribuție a aplicațiilor a evoluat de la utilizarea componentelor hardware pentru stocare și distribuție (CD, DVD, Floppy disk) la descărcarea acestora de pe site-uri publice.

Există numeroase tipuri de suporturi utilizate pentru implementarea și distribuirea unei aplicații: aplicații instalabile (native), aplicații web, aplicații web hibride, aplicații web progresive.

## **3.2 Tipuri de arhitectură a aplicațiilor**

În ultimii ani, inovațiile din domeniul tehnologiilor și metodologiilor au permis dezvoltarea și implementarea aplicațiilor software într-un mod mai rapid și mai eficient. Unele dintre cele mai semnificative tendințe în dezvoltarea software-ului este trecerea de la aplicațiile cu o arhitectură monolitică la cele bazate pe microservicii.

### **3.2.1 Arhitectura Monolitică**

Cuvântul „Monolit” în dezvoltarea software-ului se referă la o aplicație formată dintr-un singur proiect în care multiple componente și servicii sunt combinate și servite sub aceeași infrastructură de aplicație servită de o singură platformă. Astfel de aplicații deservesc, de obicei, multiple domenii de interes [10].

### **3.2.2 Arhitectura bazată pe microservicii**

Arhitectura bazată pe microservicii este o variantă a Arhitecturii Orientate pe Servicii (SOA). Deși SOA a fost disponibil începând cu anul 1998, microserviciile au fost oficial adoptate în 2012 [11]. Conceptul central din spatele microserviciilor constă în mai multe servicii mici și autonome care lucrează împreună pentru a servi aceluiași scop al unei aplicații mari; o aplicație mare și complexă este împărțită în bucăți mai mici [12], organizate după sub-domenii, fiind astfel mai ușor de întreținut.

### **3.2.3 Monoliți și microservicii; analiză comparativă**

Microserviciile sunt o alegere evidentă întrucât ușurează lucrul colaborativ și permit actualizarea incrementală a părților individuale din sistem. Deasemenea, dacă există o cerere specifică pentru un anumit domeniu al aplicației, doar serviciile responsabile de acel domeniu sunt scalate pentru a satisface nevoile consumatorului [13].

## **3.3 Descentralizare folosind Blockchain**

Încă de la apariția tehnologiei blockchain, numeroase aplicații practice au stârnit interesul general în lumea dezvoltării software. Inițial, bazându-se în principiu pe criptomonede, evoluția tehnologiei a fost spectaculoasă iar datele stocate pe blockchain pot fi folosite pentru crearea de aplicații complexe cu aplicabilitate într-o varietate de domenii. Principalele avantaje ale tehnologiei blockchain sunt trasabilitatea și imutabilitatea oferite de acesta.

### **3.3.1 Blockchain-uri private**

Spre deosebire de blockchain-urile publice, sistemele private bazate pe blockchain se bazează pe mecanisme de control stricte și trasabile.

### **3.3.2 Blockchain-uri publice**

Deși blockchain-urile publice oferă o transparență totală a datelor și oricine poate participa la validarea tranzacțiilor, dezvoltarea de aplicații ce folosesc infrastructuri publice sunt vulnerabile atacatorilor întrucât codul este expus.

### **3.3.3 Stocare descentralizata (IPFS)**

IPFS (InterPlanetary File System) este un sistem distribuit peer-to-peer de fișiere care urmărește să conecteze toate dispozitivele sub același sistem de fișiere. Cel mai utilizat sistem de distribuire de date este în continuare HTTP, dar acesta prezintă deficiențe la volum mare de date. IPFS urmărește să îmbunătățească acest sistem fără a periclita experiența utilizatorului.

### **3.3.4 Contracte Inteligente**

Contractele inteligente, fundamentale pentru aplicații pe Ethereum, sunt versiuni digitale ale contractelor tradiționale, transformând acordurile în cod auto-executabil în blockchain. Spre deosebire de contractele clasice, care necesită încredere între părți pentru execuție, contractele inteligente se execută automat odată cu îndeplinirea termenilor, eliminând nevoia de încredere reciprocă.

### **3.3.5 Mecanisme de consens**

Mecanismul de consens este una dintre componentele esențiale în lumea tehnologiei blockchain, care asigură fiabilitatea acesteia. Un mecanism de consens este, în esență, o procedură complexă utilizată de rețelele blockchain[14] pentru a aduce părți diverse și, probabil, nesigure, la un consens cu privire la veridicitatea tranzacțiilor. Acest lucru nu numai că ajută la verificarea legitimității tranzacțiilor, dar acționează, de asemenea, ca o protecție împotriva utilizării monezii de doua ori. Cele mai populare mecanisme de consens utilizate la momentul scrierii acestei lucrări sunt Proof of Work (PoW) și Proof of Stake (PoS).

### **3.3.6 Securitate și vectori de atac**

Securitatea a devenit o prioritate de top în ecosistemul cu multe schimbări al tehnologiei blockchain. Blockchain-urile sunt rezistente la multe amenințări cibernetice obișnuite

datorită caracterului lor descentralizat și criptografic inerent, însă nu sunt rezistente la toate tipurile de atacuri cunoscute.

### **3.3.7 Împuternicirea utilizatorilor și guvernarea datelor**

Într-o arhitectură descentralizată, utilizatorii au controlul datelor personale, diferențiindu-se de modelul tradițional centralizat. Într-un blockchain public, este asigurată transparența și trasabilitatea datelor.

### **3.3.8 Modele de cost**

Pentru a putea calcula diferența de cost între o arhitectură centralizată și cea de tip descentralizat, se iau în calcul mai multe aspecte cum ar fi: costurile pentru infrastructură, complexitatea dezvoltării aplicațiilor și costurile generate de stocarea informațiilor.

## **3.4 Portofele electronice**

Portofelele electronice reprezintă o schimbare de paradigmă în domeniul finanțelor digitale, deoarece permit utilizatorilor să stocheze, să tranzacționeze și să își gestioneze activele financiare în format electronic.

### **3.4.1 Tipuri de portofele electronice**

Cele mai populare soluții pentru întreținerea și accesarea unui portofel electronic sunt sub forma unei aplicații pentru telefonul mobil. Totuși, există mai multe tipuri de portofele electronice, variind de la suport hardware cât și software, fiecare aducând avantaje și riscuri.

### **3.4.2 Securitatea portofelelor electronice**

Toate portofelele electronice sunt vulnerabile pentru că acestea depind de o frază pe care utilizatorul o ține minte sau o notează numită "mnemonic". Dacă această frază intră în posesia unui utilizator rău intenționat, putem să considerăm ca portofelul electronic este compromis și toate bunurile stocate la acea adresă pot fi pierdute.

### **3.4.3 Interacțiunea cu portofelele electronice**

Comunicarea dintre portofelele blockchain și aplicațiile descentralizate (dApps) se evidențiază ca un progres cheie în lumea complexă a tehnologiei blockchain, deschizând calea pentru interfețe de utilizator mai fluide și mai simple. DApps rulează pe rețele blockchain peer-to-peer, spre deosebire de programele convenționale ce utilizează servere

centralizate. Este necesară o interfață sigură pentru ca utilizatorii să interacționeze cu aceste dApps, fie pentru a efectua o tranzacție, pentru a pune în joc jetoanele sau pentru a lua parte la alte activități legate de blockchain.

## **3.5 Web 3.0**

Web3, cunoscut și sub numele de Web 3.0, este o idee a următoarei versiuni a World Wide Web, care se concentrează pe descentralizarea datelor și o economie bazată pe token-uri [15]. Descentralizarea împuternicește schimbul de informații între perechi, eliminând intermediarii și entitățile terțe care ar putea controla datele.

### **3.5.1 Arhitectura Web 3.0**

Arhitectura Web 3.0 revoluționează internetul prin descentralizare, permițând utilizatorilor să dețină controlul asupra datelor lor personale cu ajutorul tehnologiilor blockchain și al contractelor inteligente iar legătura cu sistemele clasice se realizează fără un efort considerabil.

### **3.5.2 Aplicații Web 3.0**

Există numeroase aplicații ale tehnologiei Web3.0, majoritatea punând accent pe anonimitatea participanților și pe transparența datelor. Exemple notabile ale folosirii tehnologiei sunt: DeFi (Finanțe descentralizate), NFT, DAO (Organizații autonome descentralizate).



# Capitolul 4

## Migrarea de la aplicații instalabile native la aplicații web

În acest capitol, explorez evoluția aplicațiilor, conținutul lor și tipurile de aplicații. Mai departe, explorez provocările ridicate de migrarea aplicațiilor, iar la final concluzionez cu o analiză a performanței unei aplicații populare de editare a documentelor disponibilă atât ca pachet instalabil, cât și ca aplicație web, măsurând timpul necesar pentru a deschide documente și consumul de RAM.

### 4.1 Experiența utilizatorului și provocările tehnice

S-a demonstrat că experiența utilizatorului în timpul utilizării software-ului (UX) are o influență semnificativă asupra adopției și modului de utilizare al soluției. Pe măsură ce companiile se gândesc să-și migreze programele de la pachete instalabile la aplicații online, este esențial să se ia în considerare impactul lor asupra experienței utilizatorului (UX). Există mai multe considerații tehnice implicate în realizarea unei migrări de software de la alegerea tehnologiilor la considerente de securitate a sistemului.

#### 4.1.1 Experiența utilizatorului

Din punctul de vedere al experienței utilizatorului, provocările migrației spre aplicații online, cum ar fi dependența de conexiunea la internet și curba de învățare pentru noile interfețe, se pot aborda prin furnizarea de instruiți și resurse de suport care facilitează adaptarea utilizatorilor.

#### 4.1.2 Cerințe tehnice

Provocările tehnice ale migrării aplicațiilor în mediul web, precum limitările browserelor și interacțiunea cu hardware-ul, pot fi abordate printr-o strategie ce include dezvoltarea se-

curizată, utilizarea infrastructurii cloud sigure și optimizarea performanței, îmbunătățind astfel experiența utilizatorului.

### **4.1.3 Avantajele și dezavantajele aplicațiilor instalate de utilizator**

Aplicațiile instalate de utilizator necesită în general ca utilizatorul să dețină hardware-ul pe care este instalată aplicația. Cu toate acestea, în lumea rapid evolutivă a internetului, acesta este un dezavantaj semnificativ. Unele aplicații oferă sincronizarea online a configurațiilor și conținutului utilizatorilor. Cu toate acestea, pentru a accesa aceste informații [16], utilizatorul trebuie să instaleze aceeași aplicație pe mai multe dispozitive conectate la internet.

Avantajele constau în principal în capacitățile de funcționare a aplicației fără o conexiune la internet și izolarea acesteia față de potențiali atacatori, pe când dezavantajele se rezumă la accesul limitat la aplicație și dificultatea procesului de a interacționa cu aplicația pentru prima dată

## **4.2 Aspecte cheie în tranziția către o infrastructură descentralizată**

Tranziția de la Web2 la Web3 implică o mișcare substanțială către descentralizare, iar o analiză a business-ului care sta la baza aplicației în discuție trebuie luată în considerare înainte de a efectua tranziția. Aplicațiile concepute pentru utilizare privată în cadrul unei companii pot beneficia de o arhitectură descentralizată, deoarece natura tranzacțiilor blockchain se bazează pe imutabilitate, astfel, se poate pune în aplicare o urmărire mai bună a resurselor.

Când se trece la un blockchain public și o arhitectură descentralizată, guvernanta, securitatea, trebuie să fie luate în considerare interoperabilitatea și confidențialitatea.

## **4.3 Analiză comparativă a performanței unei aplicații web disponibilă și în format instalabil**

Pentru scopul acestui studiu, a fost aleasă o soluție software populară de editare a documentelor, deoarece este disponibilă atât sub formă de pachet instalabil, cât și de aplicație web.

Dispozitivele folosite pentru comparație:

- **Dispozitivul 1:** Apple Macbook Pro, 13 inci, CPU: Apple M2, Memorie: 32GB
- **Dispozitivul 2:** Apple Macbook Pro, 16 inci, CPU: Apple M1 Max, Memorie: 64GB

Conectivitate la internet:

- 230 Mbps, București, România

Browser:

- Google Chrome, Versiunea 111.0.5563.64 (arm64)

Realizarea de simulări multiple pentru deschiderea unui fișier folosind ambele laptopuri și ambele soluții va oferi informații valoroase despre diferența de performanță între versiunea online și versiunea instalată de utilizator a aceleiași aplicații.

Timpu până la prima redare și consumul de RAM sunt factori esențiali de luat în considerare [17], deoarece oferă informații privind experiența utilizatorului. Sistemul de cache al browserului (salvare în memorie) nu a fost dezactivat, iar aplicația instalată a fost închisă complet după fiecare deschidere a fișierului.

Studiul a fost realizat cu succes folosind două tipuri diferite de fișiere (mici și mari) - 390KB și 32MB. Compararea timpilor de încărcare și a consumului de RAM pe ambele dispozitive oferă o imagine de ansamblu bună a diferențelor.

Timpu de reacție al utilizatorilor și timpu de solicitare a rețelei au fost măsurați pentru a observa dacă versiunea online a aceleiași aplicații se încarcă mai repede comparativ cu versiunea instalată local.

Timpu petrecut deschizând fișierele a fost măsurat de la începutul procesului (dublu clic pe fișierul executabil pentru aplicația instalată nativ și clic pe link pentru aplicația web) până la momentul în care documentul devine interactiv pentru 20 de rulări consecutive folosind cele două dispozitive menționate.

Calculul îmbunătățirii relative  $RI$  a fost realizat folosind formula Ecuației 4.1, unde  $N$  reprezintă valoarea nouă și  $O$  reprezintă valoarea originală.

$$RI = \frac{N - O}{O} \quad (4.1)$$

S-a observat că, folosind aplicația instalată, timpu petrecut deschizând un fișier mic este în medie de 3,994s, în timp ce deschiderea aceluiași fișier pe versiunea web are o medie de 2,160s pe **Dispozitivul 1**. Efectuând aceleași teste pe un fișier mai mare, timpu mediu de încărcare pentru aplicația nativă a fost de 4,396s comparativ cu versiunea web, care a avut o medie de 3,032s. Acest lucru reprezintă o îmbunătățire de 45,93%, respectiv 31,03%, a timpului petrecut pe versiunea online comparativ cu aplicația instalată, așa cum se vede în Figura 4.1a. În medie, timpu a scăzut cu aproximativ 38%.

Consumul de RAM pe **Dispozitivul 1** pentru operațiunea de deschidere a celor două fișiere folosind aplicația locală a fost observat a avea o medie de 351MB pentru fișierele mici și 367MB pentru fișierele mari, în timp ce pentru versiunea web, mediile au fost de 227MB și 239MB, respectiv, ceea ce reprezintă o medie de 36% mai mică consum de RAM, așa cum se vede în Figura 4.2a.

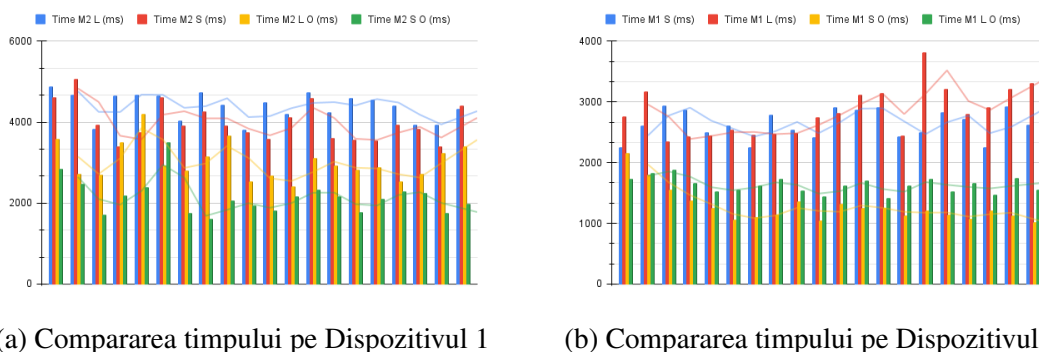


Figura 4.1 Timpul petrecut până la interactivitate

S: Fișier mic      M1: Dispozitivul 1  
L: Fișier mare      M2: Dispozitivul 2  
O: Aplicație web

Efectuând aceleași teste pe **Dispozitivul 2**, în afară de scăderea generală a timpului petrecut comparativ cu **Dispozitivul 1** (lucru de așteptat, având în vedere că specificațiile dispozitivelor sunt diferite), se poate observa că deschiderea fișierului mic pe aplicația nativă și pe versiunea web a durat în medie 2,640s și, respectiv, 1,282s. Pentru fișierul mai mare, rezultatele au fost de 2,836s și 1,630s. S-a observat o scădere medie de 51,43%, respectiv 42,5%, a timpului petrecut. În total, timpul a scăzut cu 47%. Figura 4.1b arată evoluția testelor pe dispozitivul 2.

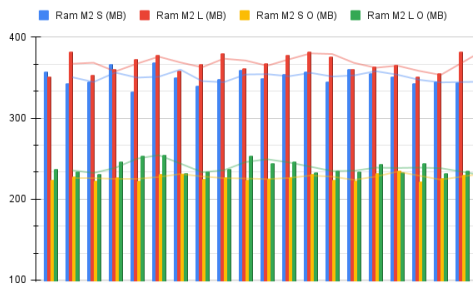
Consumul de RAM pe **Dispozitivul 2** pentru operațiunile folosind aplicația locală a fost observat a avea o medie de 250MB pentru fișierele mici și 262MB pentru fișierele mari, în timp ce pentru versiunea web, mediile au fost de 275MB și 205MB, respectiv, ceea ce reprezintă o medie de 6,25% mai mică consum de RAM, așa cum se vede în Figura 4.2b.

Timpul așteptat pentru realizarea ambelor operațiuni, folosind aplicația web și aplicația instalată local, poate varia în funcție de încărcarea rețelei, viteza internetului și încărcarea dispozitivului.

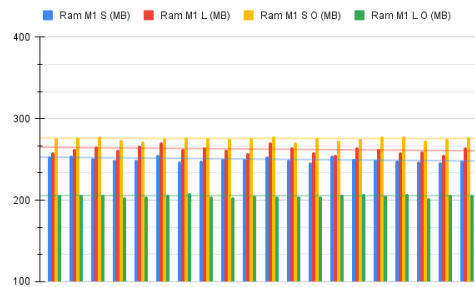
Calculând rezultatele, putem presupune că folosirea versiunii web a aceleiași aplicații necesită cu 42% mai puțin timp până când aplicația devine interactivă și consumul de RAM este cu 20% mai mic.

## 4.4 Concluzii

Pe măsură ce dezvoltarea software și internetul evoluează, este necesar să ținem pasul cu tendințele actuale. În primele etape ale dezvoltării software-ului, interacțiunea cu sistemele era simplă și se realiza în principal prin intermediul unui terminal de comandă. Tendințele recente se bazează pe interfețe utilizator moderne cu o experiență simplificată, deoarece interacțiunea cu aplicațiile de zi cu zi se mută de la aplicațiile instalate pe desktop la aplicațiile web.



(a) Compararea consumului de RAM pe Dispozitivul 1



(b) Compararea consumului de RAM pe Dispozitivul 2

Figura 4.2 Consumul de RAM  
 S: Fișier mic      M1: Dispozitivul 1  
 L: Fișier mare    M2: Dispozitivul 2  
 O: Aplicație web

# Capitolul 5

## Scalarea interfețelor grafice ale aplicațiilor web folosind micro-frontend

În acest capitol analizez soluțiile existente pentru crearea de aplicații micro-frontend și evidențiez obstacolele și avantajele aduse de acestea.

În prima secțiune, cercetez modelele actuale de arhitectură împreună cu avantajele și dezavantajele lor. Apoi, aprofundez arhitectura micro-frontend, investigând modele și tehnici[18] de implementare. În final, cercetez metodele disponibile pentru implementarea micro-frontend, fiecare cu avantaje și dezavantaje bazate pe criterii predefinite; și se face o comparație practică între trei implementări alese.

### 5.1 Micro-frontend

Micro-Frontend este o abordare de tip *Microservicii* pentru tehnologiile web front-end. Scopul principal al micro-frontend-urilor este de a împărți aplicația în mai multe unități de aplicație bazate pe bucăți de funcționalitate sau ecrane care reprezintă un anumit domeniu în loc de a crea o aplicație front-end *monolitică* reală[19].

#### 5.1.1 Tipuri de compoziție

Pentru a construi o aplicație bazată pe Micro-Frontend-uri, există câteva opțiuni diferite. De exemplu, în arhitectura bazată pe Micro-Frontend-uri, anumite decizii arhitecturale trebuie să fie luate în avans[20], deoarece aceste decizii vor influența deciziile în viitoarele implementări. Aplicațiile pot fi împărțite atât prin includerea mai multor micro-frontend-uri pe aceeași pagină cât și prin izolarea micro-frontend-urilor la unul pe pagină.

#### 5.1.2 Provocările utilizării unei arhitecturi micro-frontend

Din analiza efectuată, doar un număr limitat de studii au fost create pentru a găsi cea mai bună abordare pentru adoptarea unei arhitecturi micro-frontend. Astfel, nu

există un standard în vigoare. Beneficiile adoptării unei arhitecturi micro-frontend includ coordonarea evenimentelor eficientizată printr-un cadru robust, compatibilitatea inversă asigurată la actualizări, comunicare standardizată între componente, un model Emițător/Receptor pentru gestionarea centralizată a schimbărilor, controlul optim al dimensiunii pachetelor prin includerea selectivă a dependențelor, și consistența stilului între proiecte cu ajutorul unei biblioteci de stilizare unificate.

### 5.1.3 Beneficii

Accentuând atributele emergente și beneficiile pe care le oferă, pe lângă abordarea tehnică specifică, arhitectura micro-frontend aduce multe beneficii echipelor de dezvoltare.

### 5.1.4 Soluții Micro-Frontend

Au apărut mai multe soluții micro-frontend în ultimii ani, fiecare cu beneficii și dezavantaje. Există multiple soluții gata de utilizat prezente pe web, cum ar fi **SingleSPA** sau **NX**, dar de asemenea, multe soluții pot fi construite de la zero.

## 5.2 Analiza performanței diverselor tipuri de soluții micro-frontend

A fost utilizată o metodologie de cercetare amănunțită pentru a compara avantajele diferitelor soluții micro-frontend. Metodologia a inclus o analiză cuprinzătoare a cercetărilor și publicațiilor anterioare referitoare la subiect și analiza documentației furnizorilor terți pentru soluțiile micro-frontend existente.

Comparând avantajele și dezavantajele fiecărei soluții în ceea ce privește criteriile identificate, am determinat ce soluție ar fi mai potrivită pentru migrarea unei aplicații la o arhitectură micro-frontend.

Implementarea unei aplicații web simple a avut loc ca parte a acestui studiu, pentru a compara două soluții micro-frontend cu abordarea monolit. Pentru comparație au fost implementate variațiuni similare ale aplicației folosind iframe-uri și Module Federation. Blocurile turcoaz și mov reprezintă micro-frontend-uri pentru implementările iframe și Module federation. Bara de navigare și blocul din dreapta sus aparțin aplicației centrale folosită doar pentru încărcarea modulelor. Folosind implementarea practică, am obținut informații și concluzii referitoare la performanța fiecărei soluții micro-frontend în îndeplinirea criteriilor alese.

Analizând soluțiile implementate cu ajutorul uneltelor de inspectare oferite de către browser am putut determina și analiza atât durata de timp pentru prima desenare a conținutului, dar și durata de timp pentru încărcarea resurselor pentru fiecare tip de aplicație.

Din criteriile de evaluare enumerate în metodologia de cercetare pot fi deduse mai multe constatări privind avantajele și dezavantajele diverselor soluții analizate.

Putem calcula o îmbunătățire relativă  $RI$  folosind Ecuația 5.1, unde  $N$  reprezintă noua valoare și  $O$  reprezintă valoarea inițial observată.

$$RI = \frac{N - O}{O} \quad (5.1)$$

Analizând informațiile rezultate din implementarea unei aplicații web simple, așa cum se vede în tabelul 5.1 și figura 5.1, putem observa că soluția ce utilizează Module federation este mai performantă decât soluția implementată cu ajutorul iframe. Figura 5.1 este un grafic radar în care valorile obținute din analiză și prezentate în tabelul 5.1 sunt comparate între cele trei implementări.

Module federation arată o scădere cu 55% a timpului necesar pentru **prima desenare a conținutului** în comparație cu iframe și o creștere cu 29% a timpului necesar în comparație cu o aplicație monolit. Cu toate acestea, luând în considerare că prima desenare a conținutului depinde de dimensiunea pachetului, pentru aplicații mai mari, timpul primei desenări a conținutului va crește în comparație cu Module federation care se bazează pe încărcarea asincronă și modulele sunt încărcate pe parcurs, în funcție de nevoile utilizatorului.

Rezultatele arată că **dimensiunea pachetului** este redusă în Module federation în comparație cu soluția iframe. Duplicarea resurselor într-o soluție ce folosește iframe sau rutare este adesea inevitabilă, crescând astfel dimensiunea pachetului. Analizând rezultatele obținute și enumerate în tabelul 5.1, putem observa o scădere cu 60% a dimensiunii pachetului pentru Module federation în comparație cu implementarea iframe. Cu toate acestea, implementarea Module federation arată o creștere cu 22% a dimensiunii pachetului în comparație cu abordarea monolit.

În ceea ce privește **numărul de solicitări** și dimensiunea totală a resurselor încărcate, Module federation depășește implementarea iframe, dar este mai puțin performant decât implementarea monolit. Implementarea iframe prezintă de asemenea o creștere semnificativă a dimensiunii totale în comparație cu abordarea monolit, în principal, deoarece resursele nu sunt împărțite între diferitele părți ale aplicației. Module federation arată o scădere cu 23% a numărului de cereri în comparație cu implementarea iframe și o creștere cu 100% a numărului de cereri în comparație cu soluția monolit.

**Timpul de încărcare** se calculează pe baza încărcării inițiale a resurselor și, deoarece Module federation încarcă asincron module secundare, există o îmbunătățire semnificativă a vitezei în comparație cu monolitul și iframe. Rezultatele arată o scădere cu 30% atunci când se compară timpul de încărcare a module federation cu iframe și o scădere cu 42% atunci când se compară cu abordarea monolit.

Deși rezultatele soluției Module federation arată o lipsă de performanță pe unele capitole specifice pentru implementarea studiată, putem presupune că pentru aplicații mai mari, aceasta va depăși abordarea monolit. Folosirea surselor de cod decuplate care



Tabel 5.1 Comparație a resurselor utilizate de către fiecare soluție

Tip/Criteriu	Monolit	Iframe	Module Federa- tion
Prima desenare a conținutului	418ms	1222ms	540ms
Număr de solicitări	13	34	26
Dimensiunea resurse- lor descărcate	5.4MB	16.6MB	6.6MB
Timpul de încărcare	1.35s	1.12s	0.774s

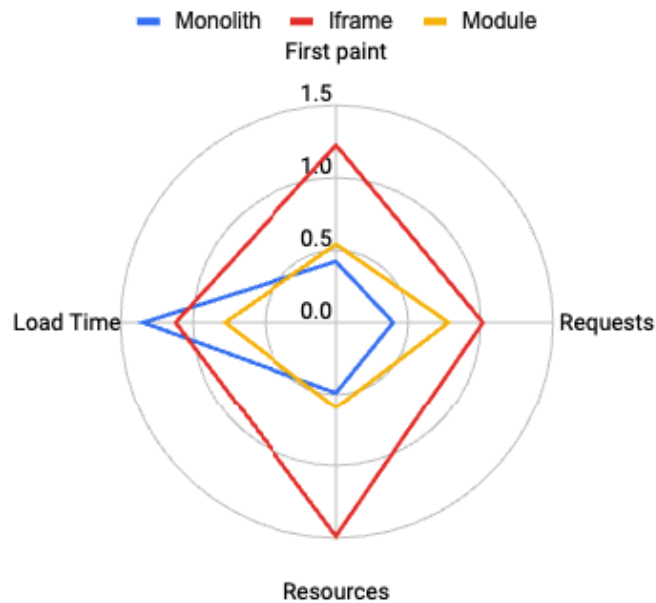


Figura 5.1 Comparație relativă între soluțiile implementate

lucrează împreună pentru a forma o întreagă aplicație oferă echipelor independență și scalabilitate fără a diminua performanța sau a degrada experiența utilizatorului.

### 5.3 Concluzii

Pe baza analizei realizate în acest capitol, se poate concluziona că micro-frontendurile oferă o soluție promițătoare pentru depășirea provocărilor aduse de aplicațiile monolitice de frontend. Prin divizarea aplicațiilor de frontend în bucăți mai mici și mai ușor de gestionat, echipele de dezvoltare pot lucra mai eficient, rezultând cicluri de dezvoltare mai scurte.

# Capitolul 6

## Web3.0 și descentralizare; aplicații practice

### 6.1 Autentificare descentralizată folosind web 3.0

În această secțiune se abordează o metodă inovatoare de autentificare bazată pe descentralizare și tehnologia blockchain Ethereum, creând legături între resursele on-chain și off-chain. Se analizează metodele actuale de autentificare, definițiile Web 3.0 și aplicațiile acestora. Este prezentată stiva tehnică propusă, avantajele și dezavantajele autentificării Web 2.0 comparativ cu Web 3.0. În final, se oferă o implementare practică a metodei propuse, cu un ghid detaliat și se discută beneficiile unui sistem anonim de autentificare. Un studiu compară timpii de autentificare ai metodei propuse cu alte mecanisme similare.

#### 6.1.1 Mecanisme de Autentificare Securizate

Mecanismul de autentificare cel mai popular se bazează pe introducerea de către utilizator a unei combinații de nume de utilizator/email și parolă pentru a obține accesul la sistem. Cu toate acestea, dacă parola este compromisă, utilizatorul ar putea pierde accesul la sistem pentru totdeauna. Există o tendință de renunțare la combinația nume de utilizator și parolă în schimbul utilizării doar a unei metode de autentificare cu un singur factor pentru a scurta timpul necesar creării unui cont pe o anumită platformă. Acest studiu își propune să exploreze viabilitatea sistemelor de autentificare bazate pe anonimare complet, astfel încât să facă o punte între operațiunile realizate prin blockchain și în afara blockchain-ului.

#### 6.1.2 Componentele sistemului

Spre deosebire de mecanismele tradiționale de autentificare, sistemul propus necesită o componentă suplimentară și anume portofelul electronic prin intermediul căruia este va-

lidată identitatea utilizatorului. Folosirea unui portofel electronic Ethereum împreună cu o aplicație tradițională web poate oferi un plus de securitate și anonimitate utilizatorilor.

### **6.1.3 Autentificarea în era Web 2.0**

Există multiple moduri de autentificare în orice aplicații bazate pe web sau instalate de utilizator, fie bazate pe acreditări ale utilizatorilor sau certificate bazându-se pe validarea secundară a identităților utilizatorilor.

Toate metodele de autentificare din Web 2.0 au un sistem centralizat de gestionare a identității care stochează acreditările utilizatorului și/sau informațiile private deținute de utilizator.

### **6.1.4 Autentificarea în era Web 3.0**

Autentificarea Web3 [21] este punctul de plecare pentru majoritatea aplicațiilor descentralizate și se bazează pe mecanismele încorporate în portofelele electronice hardware sau software, cum ar fi semnarea tranzacțiilor și mesajelor. Studiul meu se bazează pe capacitatea de semnare a mesajelor pentru a verifica faptul că utilizatorul are acces la cheia privată.

### **6.1.5 Implementarea sistemului de autentificare folosind Web3.0**

#### **Fluxul de autentificare propus**

Pentru a conecta un portofel electronic la o aplicație descentralizată, utilizatorul trebuie să confirme că site-ul poate interacționa cu portofelul, iar procesul de logare începe cu obținerea unui nonce din back-end, care apoi este semnat digital prin interfața grafică folosind adresa portofelului. După semnarea nonce-ului, mesajul este decriptat și validat de back-end, ceea ce duce la crearea unui JWT care autentifică identitatea utilizatorului pentru sesiunile viitoare pe front-end.

#### **Diagrama de autentificare**

Figura 6.1 conturează fluxul de autentificare complet folosind un software de portofel electronic terță parte.

### **6.1.6 Analiza comparativă de performanță a metodelor de autentificare**

Sistemul propus a fost implementat cu succes folosind Java și JavaScript, profitând de puterea framework-urilor Spring și Angular. Biblioteca folosită pentru a interacționa cu

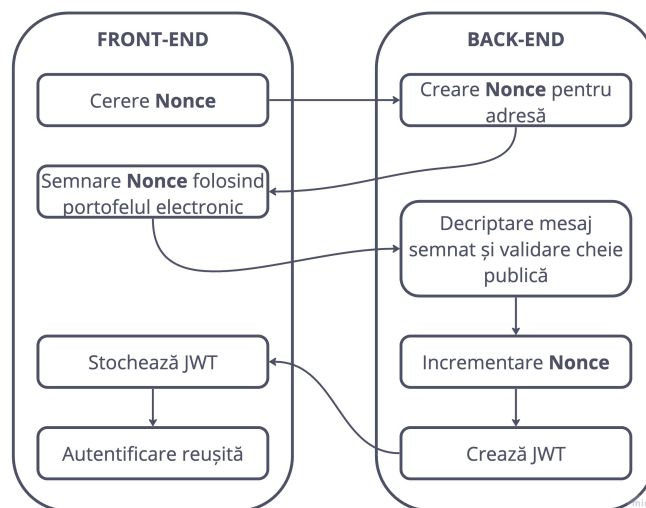


Figura 6.1 Fluxul de autentificare de la un capăt la altul.

portofelul electronic a fost Web3.js, deoarece beneficiază de sprijinul unei comunități largi și de actualizări frecvente de securitate.

Pentru a compara mecanismul de autentificare propus cu alte sisteme relevante, a fost efectuată o serie de operațiuni de autentificare (figura 6.2). Un model de autentificare relevant utilizat ca și termen de comparație în prezentul studiu este autentificarea prin SMS. Spre deosebire de 2FA tradițional, autentificarea prin SMS necesită doar ca utilizatorul să confirme autentificarea introducând codul primit prin SMS, fără a fi nevoie să introducă alte date de autentificare. S-au măsurat timpul de reacție al utilizatorilor și timpul de solicitare a rețelei pentru a observa dacă mecanismul propus are un timp de autentificare de la un capăt la celălalt mai rapid în comparație cu soluțiile similare.

După efectuarea unui număr de 50 de operațiuni consecutive ale autentificării folosind 3 metode distincte (autentificare prin portofel electronic, autentificare prin portofel electronic mobil și autentificare prin SMS), s-a observat că timpul mediu de autentificare pentru o operațiune de autentificare prin portofel electronic este de 2,698 s (Figura 6.2a). Cererea medie a rețelei este de 375 ms și poate varia în funcție de încărcarea rețelei sau a serverului. Similar cu metoda de autentificare prin portofelul virtual, timpii de solicitare pentru o operațiune de autentificare prin portofelul electronic mobil au o medie de 355 ms. Utilizând autentificarea prin portofelul electronic mobil, care depinde de un server extern pentru comunicare, a durat în medie 7,294 s (Figura 6.2b). Aceasta reprezintă o creștere cu 170% în comparație cu autentificarea standard prin portofelul electronic (2,698 s) și timpul de reacție al utilizatorilor are cel mai mare impact asupra timpului total petrecut (utilizatorul trebuie să scaneze un cod QR și să aprobe două operațiuni distincte pe telefonul mobil).

Folosind o platformă disponibilă public care a implementat un mecanism de autentificare folosind coduri SMS, a fost efectuat un număr similar de teste, iar timpul mediu petrecut în operațiune este de 13,799 s (Figura 6.2c). Încărcarea rețelei pentru acest ex-

periment este semnificativ mai mare (3,782 s) din cauza arhitecturii platformei observate. Timpul de interacțiune ai utilizatorului sunt mai mari din cauza operațiunilor care trebuie efectuate (introducerea numărului de telefon, așteptarea SMS-ului, introducerea codului SMS și așteptarea validării). În cazul autentificării folosind codul SMS (13,799s), s-a observat o creștere cu 411%, respectiv 89% comparativ cu timpul standard de autentificare prin portofelul electronic (2,698s) și timpul de autentificare prin portofelul electronic mobil (7,294s).

Timpul de așteptare pentru efectuarea unei autentificări de la un capăt la celălalt poate varia în funcție de încărcarea rețelei, timpii de reacție ai utilizatorilor și arhitectura implementată. Cu toate acestea, configurarea autentificării prin portofelul electronic propusă pare să fie mai rapidă în comparație cu alte soluții observate.

Rezultatele arată că soluția propusă are un timp total de autentificare semnificativ scăzut comparat cu metoda tradițională. Efectuarea unei autentificări utilizând un portofel electronic software de extensie a browser-ului este de cinci ori mai rapidă comparativ cu mecanismul de autentificare prin SMS observat.

### **6.1.7 Concluzii**

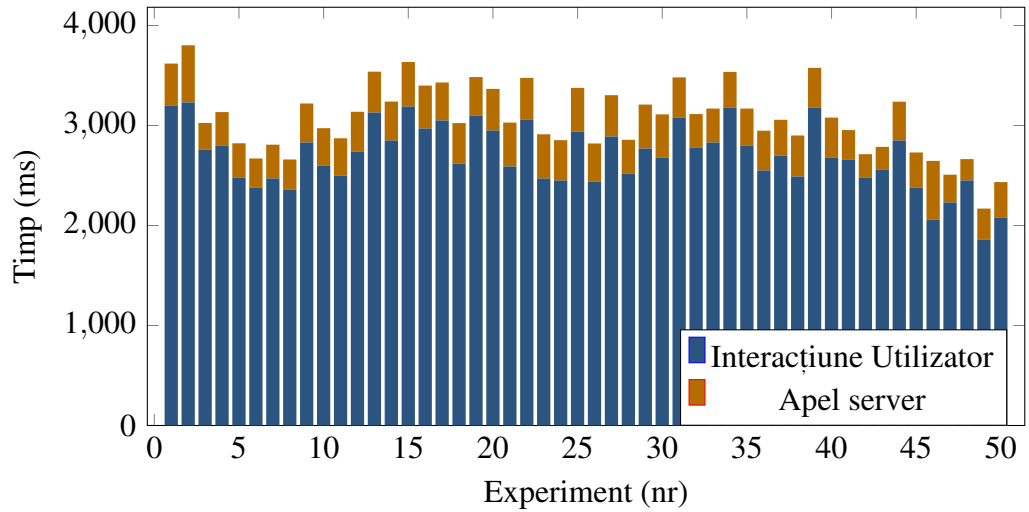
Aplicațiile descentralizate care se bazează pe calcule off-chain pot fi accesate doar utilizând o adresă de portofel electronic anonimă. Cu toate acestea, identitățile utilizatorilor care dețin anumite conturi trebuie validate printr-un proces KYC pentru operațiuni financiare mai sensibile.

## **6.2 Implementarea practică a unui sistem de semnare digitală a documentelor folosind tehnologia blockchain**

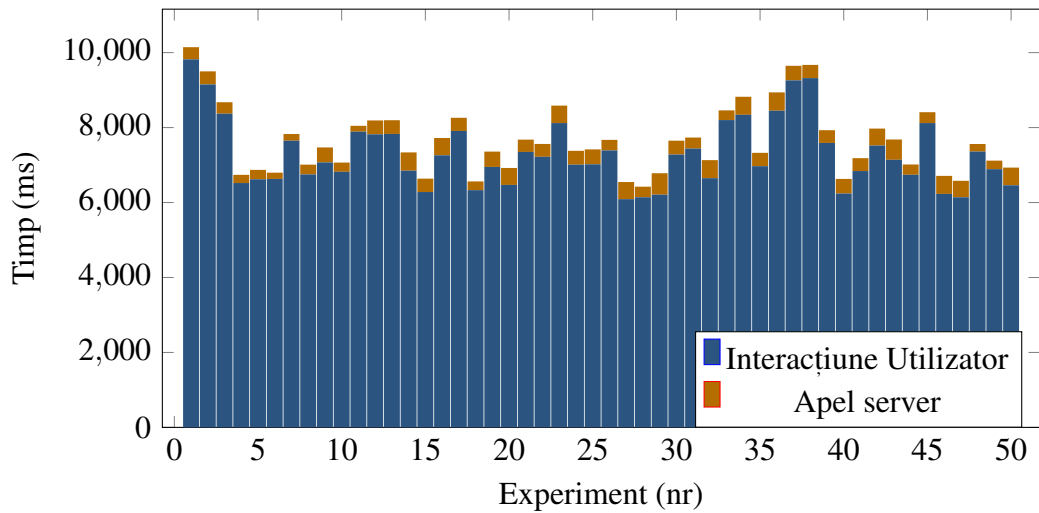
În prima parte a acestei secțiuni, voi explora tipurile de mecanisme de semnătură a documentelor, istoria lor, disponibilitatea și acceptarea lor de către guvern. Apoi, voi continua investigând tipurile de tehnici de sumă de control și de hash, urmate de secțiunea în care voi descrie o propunere de implementare practică a sistemului nostru, împreună cu avantajele și dezavantajele acesteia.

### **6.2.1 Platforme pentru semnături digitale**

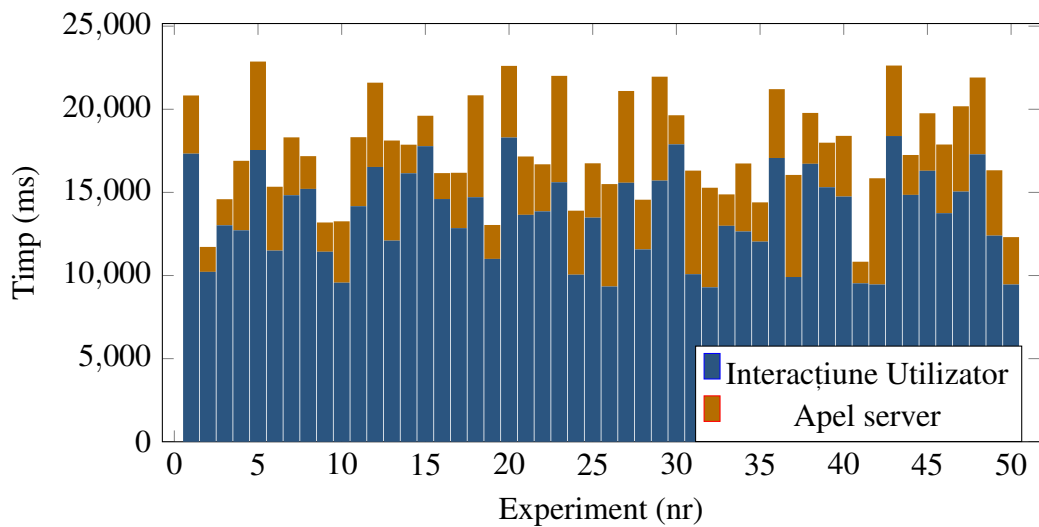
Studiul nostru vizează explorarea fezabilității sistemelor de semnătură a documentelor bazate pe blockchain, utilizând mecanismele de hash și descentralizarea. În plus, implementarea soluției propuse nu necesită modificarea documentului inițial prin atașarea semnăturilor, spre deosebire de mecanismele tradiționale de semnătură a documentelor."



(a)



(b)



(c)

Figura 6.2 Timp petrecut pentru autentificare. (a) Autentificare cu portofel electronic; (b) Autentificare cu portofel electronic mobil; (c) Autentificare cu SMS.

## 6.2.2 Tipuri de semnare a documentelor

Există mai multe tipuri de autentificare a conținutului documentelor, variind de la prezența fizică la interacțiunea digitală cu documentele.

CertIFICATELE digitale sunt emise de o terță parte de încredere, cum ar fi o autoritate de certificare (CA), și sunt utilizate pentru a verifica identitatea semnatarului [22]. Atunci când un document este semnat folosind un certificat digital, semnătura este criptată folosind cheia privată care este asociată cu certificatul. Acest lucru creează o înregistrare sigură, evidentă pentru tentativa de falsificare, a conținutului documentului la momentul semnării [23]. Destinatarul unui document semnat digital poate verifica autenticitatea și integritatea documentului utilizând cheia publică din certificat, în timp ce semnăturile biometrice și cele scrise de mână oferă metode de autentificare personală prin caracteristici unice fizice sau marcări manuale pentru a valida acordul cu conținutul documentului.

## 6.2.3 Semnături digitale

- **Hash-ul criptografic:** Această metodă implică generarea unei valori hash unice pentru un document folosind un algoritm criptografic, cum ar fi SHA-256 [24]. Valoarea hash este apoi criptată folosind cheia privată a semnatarului, creând o semnătură digitală. Semnătura poate fi verificată prin decriptarea acesteia folosind cheia publică a semnatarului și compararea rezultatului cu valoarea hash originală.

- **Certificate digitale:** După cum s-a menționat anterior, certificatele digitale pot fi utilizate pentru a verifica identitatea semnatarului și autenticitatea semnăturii. Acest lucru se face prin verificarea certificatului în funcție de o listă de CAs de încredere și verificarea faptului că acesta nu a fost revocat.

## 6.2.4 Hash-uri și sume de control

Hashing-ul este similar cu checksum (suma de control), dar este folosit în general pentru securitate, nu pentru detectarea erorilor. În dispozitivele care calculează hash-ul se introduce o intrare sau 'mesaj' și se returnează un șir de caractere de dimensiuni fixe, cunoscut sub numele de valoare hash sau rezumat de mesaj. Date fiind aceleași intrări, ar trebui să primim întotdeauna aceeași valoare hash. Rata de coliziune a unei funcții hash măsoară cât de probabil este ca două intrări diferite să producă aceeași ieșire hash.

Orice funcție hash poate experimenta o coliziune, dar probabilitatea ca acest lucru să se întâmple cu Keccak256 este extrem de mică. De exemplu, probabilitatea unei coliziuni cu Keccak256 este de aproximativ  $2^{-128}$ , care este practic zero pentru toate scopurile practice [25].

## 6.2.5 Implementarea sistemului propus

După cum s-a menționat anterior, blockchain este o tehnologie de registru digital distribuit care permite stocarea securizată și transparentă a datelor pe o rețea de calculatoare, asigurând faptul că toate datele trecute nu pot fi modificate. Prin urmare, putem presupune că utilizarea blockchain-ului pentru a citi lista semnatarilor pentru fiecare hash de document este o alternativă validă la modificarea documentelor în sine pentru a atașa semnăturile.

### Componentele sistemului de semnare a documentelor folosind blockchain

Sistemul propus este alcătuit din mai multe părți care lucrează împreună pentru a realiza procese descentralizate de semnare a documentelor. Desigur, validarea identității fiecărui individ trebuie să fie realizată manual pentru a ne asigura că persoana vizată semnează documentul.

**Hash-ul documentului** Componenta de hash a documentului în sistemului propus va fi responsabilă pentru generarea hash-urilor din documentele de intrare. Aceste documente pot avea orice format, de la .txt la .docx, .doc, .pdf [26]. Există mai multe biblioteci disponibile pe internet care vor facilita implementarea mai rapidă a oricărui algoritm de hash.

```
1 import sha3
2
3 k = sha3.keccak_256()
4 k.update(file.read())
5 h3 = k.hexdigest()
6
7 print(h3)
```

Listing 6.1 Generare hash keccak256 folosind Python

Calcularea hash-ului unui fișier specific poate fi obținută fie pe back-end-ul, fie pe front-end-ul aplicației. Pentru o experiență de utilizare mai bună și pentru a reduce puterea computațională necesară pentru generarea hash-ului, putem folosi pur și simplu o bibliotecă de front-end.

```
1 const keccak256 = require('keccak256')
2
3 console.log(keccak256('file contents').toString('hex'))
```

Listing 6.2 Generare hash keccak256 folosind JavaScript

**Blockchain** Deoarece contractele inteligente și datele de semnătură sunt publice, mai multe aplicații pot fi create pe baza sistemului[27] de bază. De exemplu, deoarece blockchain-ul Ethereum poate emite evenimente pentru operațiuni specifice, aplicațiile



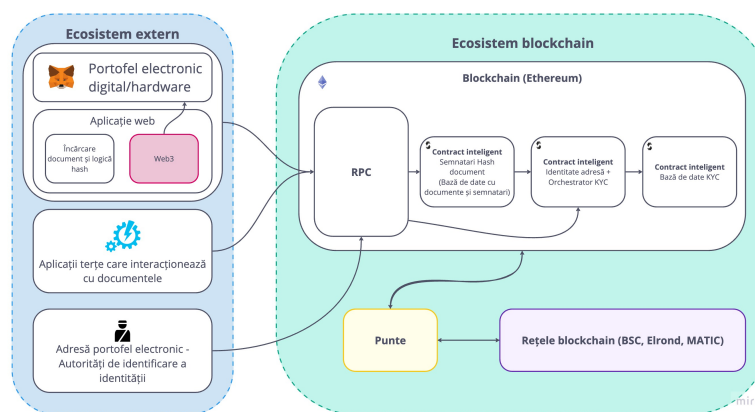


Figura 6.3 Diagrama componentelor la nivel înalt

pot asculta aceste evenimente și pot reacționa la acestea pentru afișarea datelor off-chain. Se pot adăuga clauze astfel încât, de fiecare dată când toate părțile necesare au semnat un anumit document, să poată fi executate anumite operațiuni pe blockchain.

**Aplicație web** Aplicația web este concepută pentru a fi interfața centrală și principalul punct de interacțiune cu fluxul de semnături de documente. Interacțiunea cu nodurile din rețeaua blockchain pentru a recupera informații despre starea rețelei, inclusiv tranzacții, blocuri și alte date, se realizează prin protocolul de apeluri de proceduri la distanță (RPC).

### Diagrama componentelor

După cum se prezintă în figura 6.3, numeroase componente interacționează pentru a forma sistemul complet. Deoarece sistemul este public, oricâte aplicații secundare pot fi create care ar interacționa în cele din urmă cu sistemul de bază. Un caz de utilizare ar fi condiționarea semnării unui anumit "hash" de document pentru ca un contract inteligent să fie executat.

### Fluxul de semnare și validare a documentelor prin blockchain

La fel ca în cazul proceselor de semnătură digitală disponibile, pentru a semna documentele noastre, trebuie mai întâi să calculăm un hash al documentului vizat și să îl stocăm în blockchain împreună cu adresa portofelului electronic al semnatarului. În cazul în care documentul vizat este modificat în vreun fel, hash-ul său se va schimba și, prin urmare, semnătura nu va putea fi găsită.

În figura 6.4 este prezentat un flux simplificat al creării și validării semnăturilor documentelor. Verificarea se poate face prin calcularea manuală a hash-ului documentului vizat și verificarea acestuia prin interogarea blockchain-ului sau prin utilizarea unei interfețe grafice de utilizator care calculează hash-ul și afișează toate semnăturile care au fost aplicate pe acesta, împreună cu persoana responsabilă pentru semnătură.

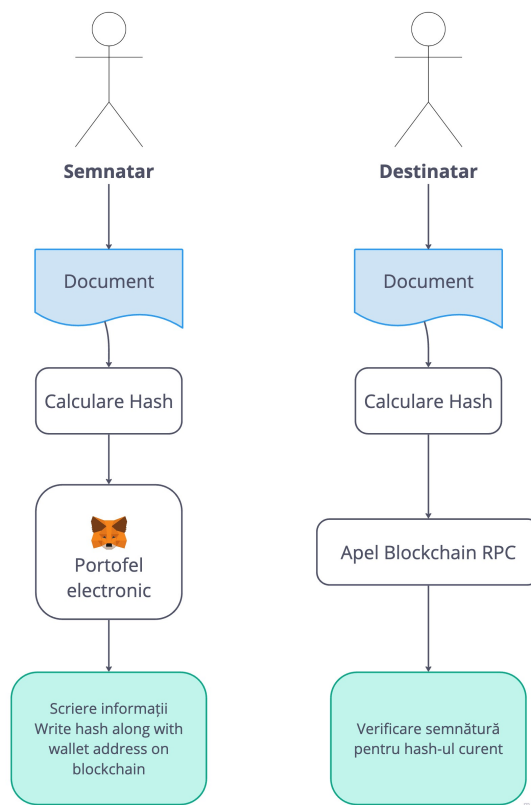


Figura 6.4 Flux complet semnare digitală a documentelor

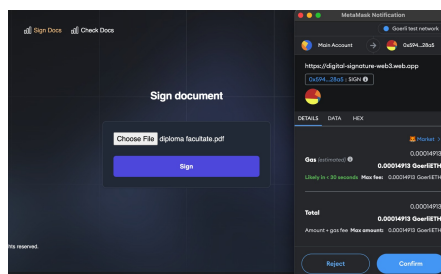


Figura 6.5 Exemplu grafic de flux semnare document

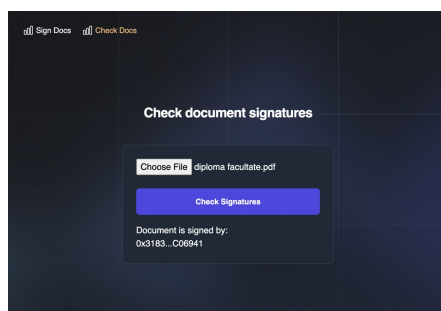


Figura 6.6 Exemplu grafic de flux de verificare semnături aplicate unui document

În figura 6.5 se poate vedea un exemplu al procesului de semnare a unui document folosind soluția propusă mai sus. Utilizatorul logat pe baza portofelului electronic încarcă un document spre semnare. Aplicația calculează hash-ul documentului folosind algoritmul keccak256 și trimite către blockchain acest hash.

Componenta de contract inteligent instalat în blockchain pentru această aplicație va prelua acest hash și îl va stoca în blockchain-ul împreună adresa portofelului electronic al utilizatorului autentificat în aplicație după confirmarea acestuia.

Blockchain-ul furnizează astfel un mediu distribuit de stocare al informației la un cost exponențial redus față de bazele de date tradiționale. În figura 6.5 se poate observa costul operației de scriere a informației în blockchain: aproximativ 0.00014913 ETH, echivalentul a aproximativ 0.27 USD, cost care este plătit de utilizatorul final și nu de aplicația care furnizează serviciul de semnare electronică de documente. Orice operație ulterioară de citire din blockchain este gratuită.

În figura 6.6 se poate vedea un exemplu de proces de verificare al semnatarilor unui document. Utilizatorul logat pe baza portofelului electronic încarcă un document spre verificare. Aplicația calculează hash-ul documentului folosind algoritmul keccak256 și trimite către blockchain acest hash.

## 6.2.6 Concluzii

Chiar dacă implementarea sistemului propus are un mare potențial, există mai multe părți ale sistemului care trebuie adresate înainte ca acesta să devină complet funcțional. Guvernele trebuie să emită identități legate de blockchain care pot fi corelate cu adrese specifice ale portofelului electronic. Fără o recunoaștere oficială a portofelelor electronice software, semnăturile nu ar avea nicio valoare. Legislația guvernamentală privind semnăturile digitale trebuie să fie implementată pentru a sprijini mecanismele legate de blockchain, cum ar fi cele propuse.

# Capitolul 7

## Concluzii

Cercetarea din această lucrare a fost dedicată studiului evoluției aplicațiilor prin prisma tranziției acestora în mediul online, explorând aspecte cum ar fi scalarea interfețelor grafice și, cu o perspectivă a viitorului, explorând consecințele tranziției într-un mediu descentralizat. Pe lângă progresele tehnologice, tranziția de la software-ul instalat nativ la platforme web sofisticate reflectă transformările în normele și așteptările societății și dezvoltarea internetului ca instrument omniprezent. Scopul final fiind acela de a oferi o examinare amănunțită a acestor schimbări, subliniind semnificația fiecărei etape a acestui proces evolutiv.

### 7.1 Obiective generale și rezultate obținute

**Tranziția de la aplicații instalabile nativ**, la aplicații bazate pe tehnologii web a fost poate un prim semn al potențialului scalabil și interconectat al unui univers digital oferit de internet. Această tranziție reprezintă, pe lângă mutarea soluțiilor tehnice pe o platformă diferită, o schimbare de mentalitate și de paradigmă în crearea de soluții software. Aplicațiile web, spre deosebire de cele instalabile nativ, au pus accentul pe omniprezență și pe disponibilitate, depășind astfel granițele geografice, limitările de hardware și anunțând un viitor în care actualizările de aspect, funcționalitate și performanță se pot întâmpla la ordinul zilei.

Mulți dintre gigantii tehnologici au făcut deja pasul către aplicații disponibile doar în format web din dorința de a atinge un număr cât mai mare de utilizatori și a avea un control mai bun asupra distribuției soluției. Comparând o astfel de aplicație, disponibilă atât în format instalabil cât și web, am observat că deși există o îmbunătățire a performanței aplicațiilor web comparate cu aplicațiile native, tranziția către o astfel de infrastructură necesită schimbări ideologice din partea companiilor.

Odată cu tranziția în mediul online, nevoia de scalare și îmbunătățire a performanței a dus pe lângă schimbări arhitecturale de optimizare și reziliență a serverului și la nevoia de **scalare a interfețelor grafice folosind micro-frontends**. Pe măsură ce aplicațiile au

crescut, a crescut și amprenta lor digitală, făcându-le astfel greu de întreținut și de extins. Conceptul de micro-frontends a apărut ca o soluție pentru această problemă, permițând astfel echipelor să despartă aplicațiile GUI de tip monolit în aplicații independente, de dimensiune mică, accelerând astfel timpii de dezvoltare și oferind o experiență de navigare consistentă pe întreaga platformă.

Deși tehnologiile ce permit scalarea interfețelor grafice sunt în plină dezvoltare, există, totuși, soluții stabile ce permit despărțirea aplicațiilor GUI în micro-frontends. Un studiu comparativ de performanță a fost efectuat și soluția cea mai performantă a constat în federația modulelor (module federation). Deși această soluție nu prezintă neajunsuri de performanță evidente, avantajele soluției pot fi observate odată cu mărirea aplicațiilor în dimensiune.

Fără o examinare amănunțită a descentralizării, însă, în special prin prisma **block-chain și a mediilor web3**, povestea aplicațiilor web ar fi incompletă. Descentralizarea este o schimbare filosofică, dar și una tehnologică. Scopul său fundamental este de a schimba paradigma unui control centralizat și autoritatea singulară, fie că acestea sunt în domeniul bancar, al stocării datelor sau chiar al găzduirii de aplicații. Pe lângă avantajele tehnologice, cum ar fi toleranța la erori și securitatea, descentralizarea are efecte pozitive, cum ar fi o mai bună confidențialitate a utilizatorilor și un acces mai echitabil la resurse. Promisiunea unui internet descentralizat, în care indivizii sunt cu adevărat proprietarii datelor și activelor lor digitale, este întărită de creșterea mediilor web3.

Este important să luăm în considerare ramificațiile mai largi ale schimbărilor, în timp ce analizăm aceste evoluții enorme. De-a lungul timpului, aplicațiile web au evoluat de la simple instrumente la extensii ale țesutului nostru social care afectează modul în care interacționăm, comunicăm, lucrăm, ne jucăm și chiar gândim. Evoluția lor dezvoltă multe despre propria noastră dezvoltare ca societate digitală, despre dificultățile, aspirațiile și căutarea neîncetată a inovației.

Ca în orice călătorie evolutivă, există însă multe riscuri și dificultăți pe parcurs. Întrebările legate de confidențialitatea datelor, de funcția gigantilor tehnologici centralizați, de efectele tehnologiilor emergente precum blockchain asupra mediului și de decalajul digital vor continua să fie în centrul atenției pe măsură ce se dezvoltă aplicațiile online.

În această lucrare am studiat mai multe aplicații practice ale mediului descentralizat folosind web3 și rezultatele arată că performanța acestora poate depăși performanța sistemelor clasice. Acest lucru, cumulat cu faptul că datele sunt securizate și transparente, întăresc convingerea că aplicațiile web3 au un loc bine stabilit în viitorul evolutiv al aplicațiilor.

## 7.2 Contribuții originale

Contribuțiile majore ale autorului în această lucrare (metodologii și concepte) sunt sumarizate și prezentate în paragrafele următoare, împărțite pe numărul capitolului. Fiecare contribuție va conține următoarele informații:

- descriere succintă a contribuției
  - informații suplimentare cu privire la aplicațiile practice ale contribuțiilor atât în lumea academică cât și în cadrul companiilor private;
  - informații cu privire la unicitatea contribuției;

### În capitolul 4

- Evaluarea tendințelor de migrare a aplicațiilor din mediul instalabil nativ în mediul online. Dezvoltarea metodologiei de comparație a performanței unor aplicații existente atât în mediul instalabil cât și online.
  - Rezultatele obținute pot fi utilizate pentru planificarea strategică a dezvoltării software: Înțelegerea trendurilor privind migrarea aplicațiilor de la mediul nativ la mediul online poate ajuta organizațiile să decidă cum și unde să investească în dezvoltarea software-ului;
  - După cunoștințele mele, nu am găsit similitudini în ceea ce privește acest studiu și alte studii din literatura de specialitate;

### În capitolul 5

- Evaluarea opțiunilor referitoare la scalarea aplicațiilor de tip interfață grafică utilizator (GUI). Recomandări cu privire la abordarea scalării și o comparație practică a performanțelor unor soluții identificate;
  - Cercetarea și implementarea unor soluții populare de scalare a aplicațiilor micro-frontend, comparația performanței între soluțiile existente de scalare și varianta clasică, monolit, a aceleiași aplicații;
  - Punerea în evidență modului în care diferite abordări de scalare afectează performanța aplicațiilor GUI poate ajuta dezvoltatorii de software să ia decizii mai bine informate cu privire la designul și implementarea lor;
  - După cunoștințele mele, nu am găsit similitudini în ceea ce privește acest studiu și alte studii din literatura de specialitate;

## În capitolul 6

- Dezvoltarea unei aplicații practice de autentificare descentralizată folosind Web 3.0 și blockchain ce este foarte utilă prin implementarea în numeroasele aplicații practice din domeniul aplicațiilor distribuite;
  - Implementarea unei soluții de autentificare folosind Web3.0 folosind tehnologiile JavaScript, Java și utilizând o gamă largă de portofele electronice;
  - Soluția propusă poate fi utilizată de către organizații ce mizează pe anonimitatea participanților;
  - Soluția propusă este populară în rândul aplicațiilor descentralizate dar, după cunoștințele mele, un studiu similar de comparare a performanței nu există în literatura de specialitate;
- Dezvoltarea unui metode inovative de semnare digitală a documentelor fără alterarea acestora folosind portofelele electronice și stocarea semnăturilor în Blockchain;
  - Studiul și crearea arhitecturii unui sistem ce permite semnarea documentelor electronice utilizând uneltele furnizate de ecosistemul Blockchain;
  - Sistemul propus poate înlocui sistemele tradiționale de semnare a documentelor digitale și poate duce la crearea de aplicații descentralizate care sunt securizate de natura imutabilă a tehnologiei Blockchain;
  - După cunoștințele mele, nu am găsit abordări similare de semnare a documentelor în literatura de specialitate;

## 7.3 Lista lucrărilor originale

1. **Petcu, Adrian**, Bogdan Pahontu, Madalin Frunzete, and Dan Alexandru Stoichescu. A secure and decentralized authentication mechanism based on web 3.0 and ethereum blockchain technology. *Applied Sciences*, Vol 13(Nr 4):Pag 2231, 2023. doi: 10.3390/app13042231. URL <https://doi.org/10.3390/app13042231>, Q2, factor impact 2.9, eISSN: 2076-3417, WOS:000938088300001 *citată de 8 lucrări până la data 01.09.2023*
2. **Petcu, Adrian**, Madalin Frunzete, and Dan Alexandru Stoichescu. Benefits, challenges, and performance analysis of a scalable web architecture based on micro-frontends. *UPB Scientific Bulletin*, Vol 85(Nr 3):Pag 319, 2023, ISSN: 2286-3540, WOS:001052259100025
3. **Petcu, Adrian**, Madalin Frunzete, and Dan Alexandru Stoichescu. Evolution of applications: From natively installed to web and decentralized. In *International Conference on Computational Science and Its Applications*, pages 253–270. Springer, 2023, 3-6 Iulie 2023, Atena

4. **Petcu, Adrian**, Madalin Frunzete, and Dan Alexandru Stoichescu. A practical implementation of a digital document signature system using blockchain. In *2023 13th International Symposium on Advanced Topics in Electrical Engineering (ATEE)*, pages 1–6. IEEE, 2023, 23-25 Martie 2023, București *citată de o lucrare până la data 01.09.2023*
5. Bogdan-Ionut Pahontu, **Petcu, Adrian**, Alexandru Predescu, Diana Andreea Arsene, and Mariana Mocanu. A blockchain approach for migrating a cyber-physical water monitoring solution to a decentralized architecture. *Water*, Vol 15(Nr 16):Pag 2874, Aug 2023. ISSN 2073-4441. doi: 10.3390/w15162874. URL <http://dx.doi.org/10.3390/w15162874>, Q2, factor impact 3.5, eISSN: 2073-4441 WOS:001056031300001

### Rapoarte de cercetare

1. Adrian Petcu. Contribuții la migrarea aplicațiilor din mediul standalone în online. In *Technical Report no 1*. University POLITEHNICA of Bucharest, 2018
2. Adrian Petcu. Evoluția aplicațiilor web: De la nativ la hibrid. In *Technical Report no 2*. University POLITEHNICA of Bucharest, 2018
3. Adrian Petcu. Micro front-ends: Micro aplicații bazate pe tehnologii front-end. In *Technical Report no 3*. University POLITEHNICA of Bucharest, 2019
4. Adrian Petcu. Aplicații web progresive. alternativă viabilă a aplicațiilor native. In *Technical Report no 4*. University POLITEHNICA of Bucharest, 2019
5. Adrian Petcu. Web assembly avantaje și provocări. In *Technical Report no 5*. University POLITEHNICA of Bucharest, 2020

## 7.4 Perspective de dezvoltare ulterioară

Dezvoltarea aplicațiilor web va avea un impact asupra unei game largi de industrii, de la afacerile private la interacțiuni sociale, de la guvernare la divertisment. Subiectele abordate în această lucrare pot fi extinse prin dezvoltarea următoarelor subiecte.

- Analiza impactului pe termen lung a transferului de costuri ale business-urilor către utilizatorii sistemului în contextul aplicațiilor **Web3.0**;
- Reglementarea: Pe măsură ce aplicațiile descentralizate web devin din ce în ce mai integrate în societățile noastre, este de așteptat să asistăm la o creștere a cadrelor legislative care să reglementeze funcționarea acestora. Acesta va cuprinde de la legi privind confidențialitatea datelor, considerații antitrust, până la reguli care să asigure un acces echitabil;



- Impactul introducerii monedelor digitale asupra business-urilor. Potențialul schimbării modelelor de cost și taxare;
- Integrarea identităților digitale în sisteme existente și dezvoltarea de organizații autonome descentralizate;
- Digitalizare activelor: Pe platformele **Web3.0**, totul, de la bunuri imobiliare la proprietate intelectuală, poate fi simbolizat sub formă digitală. Acest lucru ar putea democratiza oportunitățile de investiție, permițând proprietatea fracționată a unor active care anterior erau accesibile doar câtorva persoane selecte prin proceduri notariale complexe;

# Bibliografie

- [1] Fitri Wulandari. What is Web3 | Definition and Meaning. <https://capital.com/web3-definition>, 2023. [Online; accesat 2023-07-02].
- [2] M. Metcalf. Why Fortran? *ACM SIGPLAN Fortran Forum*, 2(1):13–14, 3 1983.
- [3] Wikipedia. Timeline of programming languages — Wikipedia, the free encyclopedia. <http://en.wikipedia.org/w/index.php?title=Timeline%20of%20programming%20languages&oldid=1164870771>, 2023. [Online; Accesat 31-07-2023].
- [4] Wikipedia. Git — Wikipedia, the free encyclopedia. <http://ro.wikipedia.org/w/index.php?title=Git&oldid=15762410>, 2023. [Online; Accesat 31-July-2023].
- [5] Jackson Reeves. Git Best Practices for Team Collaboration — dev.to. <https://dev.to/jtreeves/git-best-practices-for-team-collaboration-3bf0>, 2022. [Accesat 31-07-2023].
- [6] C. Selvaraj and S. Anand. Peer profile based trust model for P2P systems using genetic algorithm. *Peer-to-Peer Networking and Applications*, 5(1):92–103, oct 1 2011.
- [7] E. C. Bank. Digital euro. [https://www.ecb.europa.eu/paym/digital\\_euro/html/index.en.html](https://www.ecb.europa.eu/paym/digital_euro/html/index.en.html), nov 8 2022. [Online; accesat 2023-08-02].
- [8] S. Licata. Digital Dollar. <https://digitaldollarproject.org/>, jun 23 2023. [Online; accesat 2023-08-02].
- [9] Xiong Zhang, Wei T Yue, and Wendy Hui. Software piracy and bundling in the cloud-based software era. *Information Technology & People*, 32(4):1085–1122, 2019.
- [10] Anfel Selmadji, Abdelhak-Djamel Seriai, Hinde Lilia Bouziane, Rahina Oumarou Mahamane, Pascal Zaragoza, and Christophe Dony. From monolithic architecture style to microservice one based on a semi-automatic approach. In *2020 IEEE International Conference on Software Architecture (ICSA)*, pages 157–168. IEEE, 2020.
- [11] Francisco Ponce, Gastón Márquez, and Hernán Astudillo. Migrating from monolithic architecture to microservices: A rapid review. In *2019 38th International Conference of the Chilean Computer Science Society (SCCC)*, pages 1–7. IEEE, 2019.
- [12] Nicola Dragoni, Saverio Giallorenzo, Alberto L Lafuente, Manuel Mazzara, Fabrizio Montesi, Ruslan Mustafin, Lara Safina, and Gianluigi Zavattaro. Microservices: yesterday, today, and tomorrow. *Communications of the ACM*, 60(6):36–44, 2017.

- [13] Calin CONSTANTINOV, Lucian IORDACHE, Adrian GEORGESCU, Paul-Stefan POPESCU, and Mihai MOCANU. Performing social data analysis with neo4j: Workforce trends & corporate information leakage. In *2018 22nd International Conference on System Theory, Control and Computing (ICSTCC)*, pages 403–406, 2018. doi: 10.1109/ICSTCC.2018.8540645.
- [14] Shubhani Aggarwal and Neeraj Kumar. Chapter eleven - cryptographic consensus mechanisms - introduction to blockchain. In Shubhani Aggarwal, Neeraj Kumar, and Pethuru Raj, editors, *The Blockchain Technology for Secure and Smart Applications across Industry Verticals*, volume 121 of *Advances in Computers*, pages 211–226. Elsevier, 2021. doi: <https://doi.org/10.1016/bs.adcom.2020.08.011>. URL <https://www.sciencedirect.com/science/article/pii/S0065245820300668>.
- [15] Ahto Buldas, Dirk Draheim, Mike Gault, Risto Laanoja, Takehiko Nagumo, Märt Saarepera, Syed Attique Shah, Joosep Simm, Jamie Steiner, Tanel Tammet, and Ahto Truu. An ultra-scalable blockchain platform for universal asset tokenization: Design and implementation. *IEEE Access*, 10:77284–77322, 2022. doi: 10.1109/ACCESS.2022.3192837.
- [16] Mihai Liviu Despa. Comparative study on software development methodologies. *Database Systems Journal*, 5(3):37–56, 2014.
- [17] Samuel Kounev, Klaus-Dieter Lange, and Jóakim von Kistowski. *Systems benchmarking: for scientists and engineers*, volume 1. Springer, 2020.
- [18] A. Patwardhan. How to scale Frontend apps using Micro Frontends | WealthDesk. <https://wealthdesk.in/blog/scaling-frontend-apps-using-micro-frontends/>, may 6 2022. [Online; accesat 2023-06-21].
- [19] Andrey Pavlenko, Nursultan Askarbekuly, Swati Megha, and Manuel Mazzara. Micro-frontends: application of microservices to web front-ends. *J. Internet Serv. Inf. Secur.*, 10(2):49–66, 2020.
- [20] Luca Mezzalana. *Building Micro-Frontends*. " O'Reilly Media, Inc.", 2021.
- [21] Zhuotao Liu, Yangxi Xiang, Jian Shi, Peng Gao, Haoyu Wang, Xusheng Xiao, Bihan Wen, Qi Li, and Yih-Chun Hu. Make web3.0 connected. *IEEE Transactions on Dependable and Secure Computing*, 19(5):2965–2981, 2022. doi: 10.1109/TDSC.2021.3079315.
- [22] Zhen Qin, Chen Yuan, Yilei Wang, and Hu Xiong. On the security of two identity-based signature schemes based on pairings. *Information Processing Letters*, 116(6): 416–418, 2016. ISSN 0020-0190. doi: <https://doi.org/10.1016/j.ipl.2016.02.003>. URL <https://www.sciencedirect.com/science/article/pii/S0020019016300096>.
- [23] Qiuxia Zhang, Zhan Li, and Chao Song. The improvement of digital signature algorithm based on elliptic curve cryptography. In *2011 2nd International Conference on Artificial Intelligence, Management Science and Electronic Commerce (AIMSEC)*, pages 1689–1691, 2011. doi: 10.1109/AIMSEC.2011.6010590.
- [24] Matthieu Finiasz and Nicolas Sendrier. Security bounds for the design of code-based cryptosystems. In *Advances in Cryptology—ASIACRYPT 2009: 15th International Conference on the Theory and Application of Cryptology and Information Security, Tokyo, Japan, December 6–10, 2009. Proceedings 15*, pages 88–105. Springer, 2009.

- [25] Shahram Bakhtiari, Reihaneh Safavi-Naini, Josef Pieprzyk, et al. Cryptographic hash functions: A survey. Technical report, Citeseer, 1995.
- [26] J. Zhang, Y. Wang, and X. Liu. Blockchain-based digital signature: A survey. *IEEE Communications Surveys and Tutorials*, 2020.
- [27] Weidong Fang, Wei Chen, Wuxiong Zhang, Jun Pei, Weiwei Gao, and Guohui Wang. Digital signature scheme for information non-repudiation in blockchain: a state of the art review. *EURASIP Journal on Wireless Communications and Networking*, 2020(1):1–15, 2020.
- [28] **Petcu, Adrian**, Bogdan Pahontu, Madalin Frunzete, and Dan Alexandru Stoichescu. A secure and decentralized authentication mechanism based on web 3.0 and ethereum blockchain technology. *Applied Sciences*, Vol 13(Nr 4):Pag 2231, 2023. doi: 10.3390/app13042231. URL <https://doi.org/10.3390/app13042231>.
- [29] **Petcu, Adrian**, Madalin Frunzete, and Dan Alexandru Stoichescu. Benefits, challenges, and performance analysis of a scalable web architecture based on micro-frontends. *UPB Scientific Bulletin*, Vol 85(Nr 3):Pag 319, 2023.
- [30] **Petcu, Adrian**, Madalin Frunzete, and Dan Alexandru Stoichescu. Evolution of applications: From natively installed to web and decentralized. In *International Conference on Computational Science and Its Applications*, pages 253–270. Springer, 2023.
- [31] **Petcu, Adrian**, Madalin Frunzete, and Dan Alexandru Stoichescu. A practical implementation of a digital document signature system using blockchain. In *2023 13th International Symposium on Advanced Topics in Electrical Engineering (ATEE)*, pages 1–6. IEEE, 2023.
- [32] Bogdan-Ionut Pahontu, **Petcu, Adrian**, Alexandru Predescu, Diana Andreea Arsenescu, and Mariana Mocanu. A blockchain approach for migrating a cyber-physical water monitoring solution to a decentralized architecture. *Water*, Vol 15(Nr 16):Pag 2874, Aug 2023. ISSN 2073-4441. doi: 10.3390/w15162874. URL <http://dx.doi.org/10.3390/w15162874>.
- [33] Adrian Petcu. Contribuții la migrarea aplicațiilor din mediul standalone în online. In *Technical Report no 1*. University POLITEHNICA of Bucharest, 2018.
- [34] Adrian Petcu. Evoluția aplicațiilor web: De la nativ la hibrid. In *Technical Report no 2*. University POLITEHNICA of Bucharest, 2018.
- [35] Adrian Petcu. Micro front-ends: Micro aplicații bazate pe tehnologii front-end. In *Technical Report no 3*. University POLITEHNICA of Bucharest, 2019.
- [36] Adrian Petcu. Aplicații web progresive. alternativă viabilă a aplicațiilor native. In *Technical Report no 4*. University POLITEHNICA of Bucharest, 2019.
- [37] Adrian Petcu. Web assembly avantaje și provocări. In *Technical Report no 5*. University POLITEHNICA of Bucharest, 2020.