

UNIVERSITATEA “ALEXANDRU IOAN CUZA” DIN IAȘI

**FACULTATEA DE INFORMATICĂ**



**LUCRARE DE LICENȚĂ**

Proofchain - Trasabilitate folosind blockchain și contracte  
inteligente

**propusă de**

Alexandru Cambose

Sesiunea: *Iulie, 2021*

**Coordonator științific**

Lect. Dr. Panu Andrei

# Proofchain - Trasabilitate folosind blockchain și contracte inteligente

Alexandru Cambose

Sesiunea: *Iulie, 2021*

**Coordonator științific**

Lect. Dr. Panu Andrei

Avizat,

Îndrumător Lucrare de Licență

Titlul, Numele și prenumele \_\_\_\_\_

Data \_\_\_\_\_ Semnătura \_\_\_\_\_

**DECLARAȚIE privind originalitatea conținutului lucrării de licență**

Subsemnatul(a)

domiciliul în

născut(ă) la data de ....., identificat prin CNP ....., absolvent(a) al(a) Universității „Alexandru Ioan Cuza” din Iași, Facultatea de ..... specializarea ..... declar pe propria răspundere, cunoscând consecințele falsului în declarații în sensul art. 326 din Noul Cod Penal și dispozițiile Legii Educației Naționale nr. 1/2011 art.143 al. 4 si 5 referitoare la plagiat, că lucrarea de licență cu titlul:

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

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

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

Dată azi, ..... Semnătură student .....

## DECLARAȚIE DE CONSUMĂMÂNT

Prin prezenta declar că sunt de acord ca Lucrarea de licență cu titlul „*Proofchain - Trasabilitate folosind blockchain și contracte inteligente*”, codul sursă al programelor și celealte 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, data

Absolvent Prenume Nume

---

(semnătura în original)

# Cuprins

<b>Introducere</b>	<b>7</b>
<b>Contribuții</b>	<b>13</b>
<b>Capitolul 1 - Descrierea aplicației Proofchain</b>	<b>14</b>
1.1 Problema adresată	14
1.2 Soluția propusă	14
1.3 Cerințe	14
1.4 Funcționalitățile aplicației	16
1.4.1 Introducere	16
1.4.2 Autentificarea	20
1.4.3 Crearea unei identități	23
1.4.4 Administrarea materialelor	23
1.4.5 Administrarea loturilor	28
1.4.6 Administrarea transporturilor	29
1.4.7 Administrarea certificateelor	31
1.4.8 Vizualizarea istoricului unui produs	33
1.5 Integrarea cu soluțiile existente	37
1.6 Documentația utilizatorilor	37
1.7 Aplicația prezentatională	38
1.8 Soluții similare	39
1.8.1 Provenance	39
1.8.2 Everledger	39
1.8.3 OriginTrail	40
1.8.4 ShipChain	40
1.8.5 Hyperledger Fabric	40
<b>Capitolul 2 - Arhitectura aplicației</b>	<b>41</b>
2.1 Contracte inteligente - Smart Contracts	41
2.1.1 Arhitectura generală	41
2.1.2 Autentificare	42
2.1.3 Contracte	42
2.1.3.1 Companii	42
2.1.3.2 Materiale	44
2.1.3.3 Loturi	45
2.1.3.4 Transport	46
2.1.3.5 Autoritatea de certificare	47
2.1.3.6 Certificate	47
2.1.4 Testarea contractelor	48
2.1.5 Încărcarea contractelor	48
2.2 Proofchain.js Library - Librăria pentru integrare	48
2.2.1 Instalare	49

2.2.2 Inițializare	49
2.2.3 Integrarea cu Web3.js	49
2.2.4 Integrarea cu contractele inteligente	50
2.2.5 Funcționalități specifice contractelor	50
2.2.6 ABI (Application Binary Interface)	50
2.2.7 Modificarea costului tranzacției	51
2.2.8 Testarea librăriei	52
2.2.9 Documentarea librăriei	52
2.3 Proofchain Web Dashboard	53
2.3.1 Autentificarea	53
2.3.2 Comunicarea cu blockchain	54
2.3.3 Administrarea stării	55
2.4 Proofchain Web Client	55
<b>Capitolul 3 - Scenarii de utilizare</b>	<b>56</b>
3.1 Crearea identității digitale ale unei companii	56
3.2 Crearea unui material compus	57
3.3 Crearea unei instanțe a unui material compus	59
3.4 Crearea unui transport către o altă entitate	60
3.5 Vizualizarea istoricului unui material de către consumator	61
<b>Capitolul 4 - Concluzii și îmbunătățiri viitoare</b>	<b>62</b>
<b>Bibliografie</b>	<b>63</b>
<b>Anexa 1 - Arhitectura Blockchain</b>	<b>67</b>
1.1 Istorie	67
1.2 Arhitectura	68
1.3 Tipuri de blockchain	72
1.4 Utilizari în industrie	73
<b>Anexa 2 - Platforma Ethereum</b>	<b>75</b>
2.1 Introducere	75
2.2 Arhitectura	75
<b>Anexa 3 - Diagrame UML</b>	<b>80</b>

# Introducere

O revoluție industrială este considerată o perioadă de tranziție majoră către noi procese de fabricație, susținută de progresele în știință și tehnologie. Pe parcursul existenței omenirii, s-au conturat patru revoluții industriale ce au schimbat fundamental lumea în care trăim astăzi. Prima revoluție este considerată ca fiind de o importanță deosebită pentru umanitate [1], aceasta a avut loc la sfârșitul secolului al 18-lea și a constat în mecanizarea proceselor și a agriculturii, precum și inventia locomotivei cu aburi.

A doua revoluție industrială a început la sfârșitul secolului al 19-lea și a fost cunoscută datorită proceselor de producție a materialelor sintetice, precum și crearea unor linii de asamblare automate.

Incepând cu anii 1950, a treia revoluție industrială, apariția semiconducitorilor și a sistemelor de calcul a facilitat digitalizarea comunicării și robotizarea proceselor de fabricație.

Transformările de astăzi nu reprezintă doar o prelungire a celei de-a treia revoluții industriale, ci mai degrabă sosirea unei a patra, prin viteza, întinderea și impactul acestiei asupra fiecărei industrii. Robotica, inteligența artificială, internet of things (IoT), imprimanta 3D, vehiculele autonome, nanotehnologia, biotehnologia, blockchain sunt câteva dintre tehnologiile ce au deja un impact major asupra vieții cotidiene [2].

Conform unui studiu efectuat de McKinsey Global Institute (2013), 12 tehnologii revoluționare, prezентate în Figura 1, vor schimba semnificativ situația de pe piața globală până în anul 2025 [3], iar potențialul impact economic ar fi între 14 și 33 de trilioane de dolari.

Odată cu aceste evoluții tehnologice, modurile de fabricație ale diferitelor produse sunt din ce în ce mai complexe, iar lanțurile de aprovizionare evoluează în rețele de lanțuri de aprovizionare. Pentru a reuși în contextul unei economii într-o continuă digitalizare, corporațiile trebuie să gestioneze resursele de tip logistic, uman și tehnologic nu doar în interiorul întreprinderii, ci și în exteriorul acesteia, cu furnizorii, clienții și partenerii de afaceri [4]. Obiectivul unui lanț de aprovizionare fiind de a maximiza valoarea adăugată per total, nu doar în cadrul unui segment a acestuia.

Evoluția tehnologică a facilitat îmbunătățirea traiului de zi cu zi, iar oamenii au început din ce în ce mai mult să fie interesati asupra provenienței și al impactului produselor pe care le achiziționeaza. De asemenea, transparenta în cadrul lanțului de aprovizionare al alimentelor a devenit o responsabilitate impusă din punct de vedere legislativ de Uniunea Europeană în Ianuarie 2005 [5]. Conform unui studiu efectuat în peste 70 de companii din 26 de țări, aproximativ 92% dintre consumatori sunt interesați de impactul social, asupra

	<b>Mobile Internet</b>	Increasingly inexpensive and capable mobile computing devices and Internet connectivity
	<b>Automation of knowledge work</b>	Intelligent software systems that can perform knowledge work tasks involving unstructured commands and subtle judgments
	<b>The Internet of Things</b>	Networks of low-cost sensors and actuators for data collection, monitoring, decision making, and process optimization
	<b>Cloud technology</b>	Use of computer hardware and software resources delivered over a network or the Internet, often as a service
	<b>Advanced robotics</b>	Increasingly capable robots with enhanced senses, dexterity, and intelligence used to automate tasks or augment humans
	<b>Autonomous and near-autonomous vehicles</b>	Vehicles that can navigate and operate with reduced or no human intervention
	<b>Next-generation genomics</b>	Fast, low-cost gene sequencing, advanced big data analytics, and synthetic biology ("writing" DNA)
	<b>Energy storage</b>	Devices or systems that store energy for later use, including batteries
	<b>3D printing</b>	Additive manufacturing techniques to create objects by printing layers of material based on digital models
	<b>Advanced materials</b>	Materials designed to have superior characteristics (e.g., strength, weight, conductivity) or functionality
	<b>Advanced oil and gas exploration and recovery</b>	Exploration and recovery techniques that make extraction of unconventional oil and gas economical
	<b>Renewable energy</b>	Generation of electricity from renewable sources with reduced harmful climate impact

Figura 1: 12 tehnologii revoluționare. [3]

mediului, sănătății și siguranței produselor, 90% sunt interesați de transparentă în legătură cu aceste probleme mai mult decat erau în urma cu 5 ani, iar 95% consideră că interesul

consumatorilor asupra transparentei va crește pe viitor [6]. Acest fapt dovedește nivelul scăzut de încredere al consumatorilor în companii, iar asigurarea transparentei chiar și asupra unor probleme poate avea un efect benefic pe termen lung.

De exemplu, în cazul industriei alimentare, este extrem de important ca procesul de fabricație al alimentelor să fie realizat cu un grad mare de transparență. Trasabilitatea trebuie să poată fi efectuată bidirectional, atât de la materialele de bază la produsul final, cât și de la produsul final la materialele de bază, precum în Figura 2.

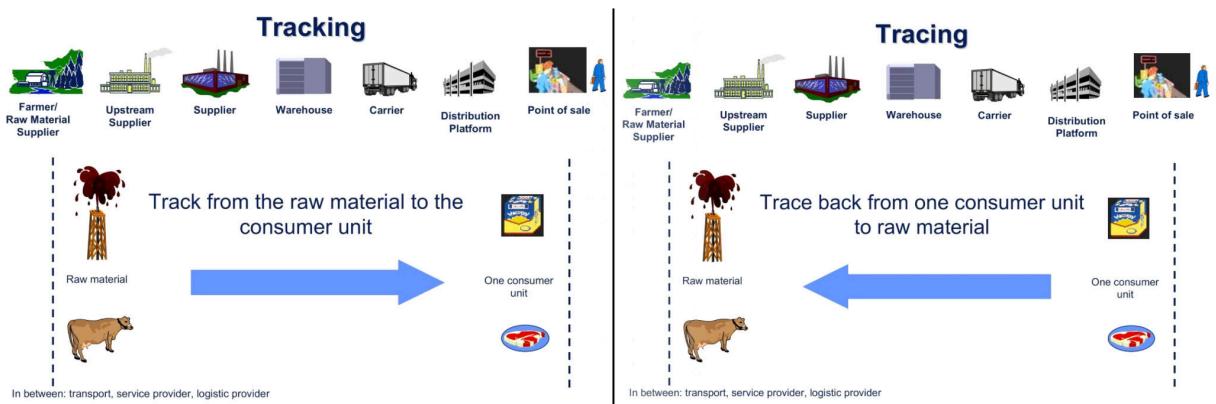


Figura 2: Trasabilitatea alimentelor [7]

Numeroase industrii și organizații investesc în cercetare și experimentează diferite tehnologii ce facilitează implementarea unui mod de operare al lanțului de aprovizionare pentru a mari transparenta. Unele dintre cele mai mari provocări ale acestui scop sunt următoarele [8]:

- *Comunicarea cu un număr mare de entități interdependente.* Companiile mari operează cu mulți actori precum transportatori, producători, vânzători și consumatori. Aceste entități nu pot stabili întotdeauna un nivel de încredere ridicat;
- *Stocarea datelor poate fi dificilă și costisitoare.* Odată cu evoluția internetului, multe dintre entități se extind în diferite țări. Deși odată cu a treia revoluție industrială, s-au schimbat radical procesele de producție, modul de a ține contabilitatea asupra operațiilor de cumpărare și vânzare încă este bazat pe documente și dosare fizice. În cazul în care compania este nevoită să furnizeze informații asupra unor tranzacții complexe sau mai vechi, accesarea acestora este una consumatoare de timp și costisitoare;
- *Securitatea și încrederea.* Sisteme existente de management al lanțului de aprovizionare sunt centralizate, fapt ce duce la o vulnerabilitate din punct de vedere al integrității datelor.

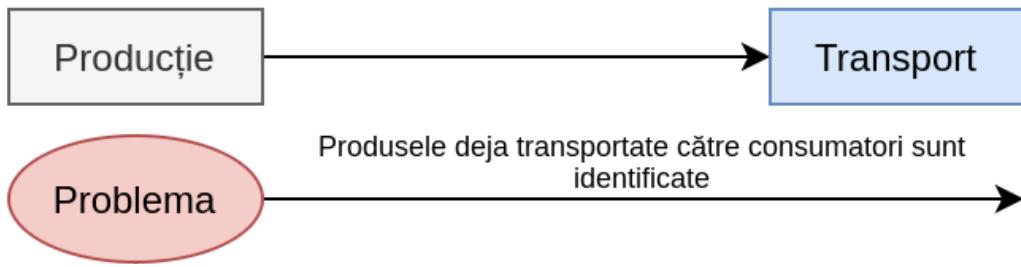
Toate aspectele menționate mai sus duc la concluzia ca problema trasabilității produselor este una tot mai complicată, dar cu un mare potențial benefic atât pentru consumatori, pentru companii dar și pentru viitorul planetei.

Putem spune că transparenta unui lanț de aprovizionare este o problemă generică, în funcție de tipul de produs și procesul de fabricație al acestuia, ce are la baza abilitatea părților rețelei de a urmări un lot de produse și istoria sa, din momentul în care acesta este creat până când ajunge să fie stocat, transportat, procesat și distribuit [9]. Aceste informații trebuie să fie disponibile și stocate fără pierderi, zgromot, întârzieri și distorsiuni.

Câteva dintre obiectivele unui sistem de trasabilitate pe întreg lanțul de aprovizionare sunt:

- *Sigur și securizat.* Fiecare tranzacție și operație trebuie să fie stocată și imposibilă de modificat;
- *Capacitatea sistemului de a crea o identitate digitală a companiilor, produselor și a loturilor de produse.* Corelarea dintre un produs fizic și omologul său digital este esențială. Acest lucru se poate face prin atașarea de diferite etichete de tipul RFID, NFC, numere serial sau coduri QR;
- *Capacitatea sistemului de a defini procese, transformări și transporturi.* Pentru a oferi o vizibilitate sporită, toate modificările și schimbările de produse trebuie înregistrate. Acest lucru este unul dintre cele mai importante obiective ale unui sistem de trasabilitate. Comisia Europeană introduce conceptul “De la ferma la consumator” (“From farm to fork”) ce încurajează utilizarea unor practici transparente și sustenabile în industria alimentară [10]. Aceste lucruri susțin importanța deosebită a unei trasabilitati bidirectionale, precum în Figura 3;
- *Integrare cu soluțiile actuale de management.* Având în vedere complexitatea programelor actuale de management al lanțului de aprovizionare, costurile de dezvoltare ale unui nou sistem ar putea fi prea ridicate și irealizabile. De aceea, la momentul actual, raportul cost/beneficii ar putea fi favorabil pentru un construirea unui sistem ce poate fi adoptat într-un mod incremental;
- *Respectarea reglementărilor și certificare.* Normele la nivel guvernamental, dar și consumatorii sunt interesați nu doar de proveniența produselor cumpărate ci și de sustenabilitatea procesului de fabricație. Cu ajutorul unor certificări, companiile își pot exprima practicile și impactul asupra mediului nu doar pentru a se conforma reglementarilor, ci și pentru a stabili un nivel ridicat de încredere față de consumatori și de parteneri;
- *Ușor de accesat pentru consumatori.* Datele despre istoricul produselor trebuie să poată fi accesate și înțelese cu usurință de către oricine.

## Transabilitate înainte (Forward tracing)



## Transabilitate înapoi (Backward tracing)

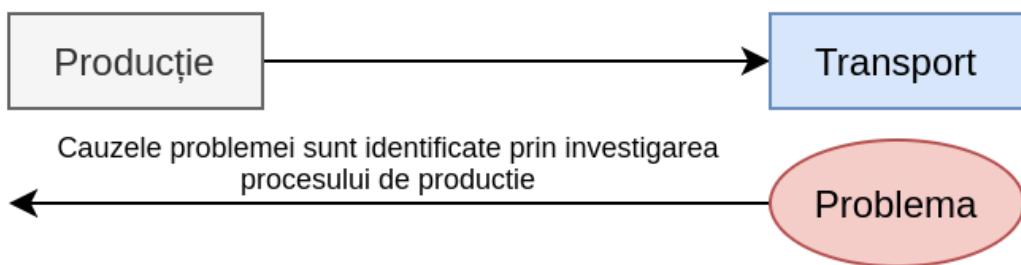


Figura 3: Tipuri de trasabilitate

Un rol important în alegerea acestui proiect este studiul tehnologiei blockchain. Similar cu începuturile internetului în anii 1970, tehnologia blockchain este tot mai răspândită, iar înțelegerea modului de funcționare al acesteia este benefic în dezvoltarea unor tipuri de aplicații. Conform unui studiu realizat de Deloitte în anul 2020 [11] pe un eșantion de 1488 de membri ai comitetului de conducere a unor companii din 14 țări, aproximativ 55% din participanți consideră că în următorii 2 ani, blockchain va fi o prioritate de top din punct de vedere al relevanței, iar 88% consideră că această tehnologie va atinge adoptia la nivel global. Din perspectiva utilizării în domeniul finanțier, aproximativ 83% din cei 1488 de participanți au considerat că activele digitale vor fi o alternativă sau vor înlocui complet monedele fiduciare (bancnotele).

De asemenea, atenția consumatorilor, inclusiv din punct de vedere personal, a crescut semnificativ asupra provenienței și a impactului produselor achiziționate. Trasabilitatea a devenit, prin urmare, esențială pentru binele comun a întregii societăți.

### Structura lucrării

În continuare, prezenta lucrare conține detaliile de implementare a unei soluții de trasabilitate precum și o evidențiere a tehnologiilor folosite pentru realizarea acesteia. Capitolul **Descrierea aplicației Proofchain** prezintă o analiză a soluției propuse în contextul problemei trasabilității, principalele funcționalități ale aplicației și o scurta menționare a soluțiilor deja existente pe piață. Capitolul **Arhitectura aplicației** cuprinde o analiză a

modului de construcție a aplicației, a abordărilor și conceptelor folosite în implementarea acesteia alături de motivația utilizării acestor abordări. Capitolul **Scenarii de utilizare** evidențiază câteva dintre cele mai importante scenarii posibile, referitoare la comunicarea dintre sistem și actorii externi. Capitolul **Concluzii** prezintă concluziile deduse în urma lucrului la aceasta aplicatie, posibilele direcții de dezvoltare pentru viitor, dar și îmbunătățirea celor existente.

## Contribuții

În contextul elaborării lucrării de licență, și motivat de studiul problemelor existente în piață, am conceput o soluție de trasabilitate în contextul unui lanț de aprovizionare, cu obiectivul de a ușura cat mai mult integrarea acestuia într-un context practic. Grupurile țintă pe care aceasta soluție dorește să le atingă sunt atât entitățile de pe lanțurile de aprovizionare cât și consumatorii finali. Aceasta este împărțită pe mai multe componente, printre care se numără: o serie de contracte inteligente, o aplicație web destinată entităților din lanțul de aprovizionare, o aplicație web destinată consumatorilor finali ale produselor, o librărie de integrare, două situri web care documentează soluția propusă și un sit web pentru prezentarea acesteia. Arhitectura este concepută având ca repere bunele practici din industrie în legătură cu dezvoltarea unui astfel de sistem, cât și studiul unor sisteme deja existente și ale unor articole științifice.

# Capitolul 1 - Descrierea aplicației Proofchain

## 1.1 Problema adresată

Trasabilitatea lanțului global de aprovisionare a devenit o problemă din ce în ce mai importantă în ultimii ani. Guvernul, mass-media, companiile, furnizorii și clienții sunt cu toții interesați de o transparență mai mare asupra întregului proces de fabricație și distribuție.

## 1.2 Soluția propusă

Soluția pe care o vom propune este “Proofchain - Trasabilitate folosind blockchain și contracte inteligente” ce își dorește să vină în întâmpinarea entităților din lanțurile de aprovisionare și a consumatorilor. Aceasta poate fi utilizată de sine stătătoare sau integrată în sistemele existente de management a inventarului și permite companiilor să înregistreze fiecare acțiune asupra unui produs. De asemenea, consumatorii vor putea vedea cu precizie proveniența fiecărui produs. Această soluție este compusă din mai multe componente precum:

- Proofchain Smart Contracts - o serie de contracte inteligente ce realizează arhitectura de baza a soluției;
- Proofchain Web Dashboard - o aplicație web destinată entităților din lanțul de aprovisionare;
- Proofchain Web Client - o aplicație web destinată consumatorilor finali ale produselor;
- Proofchain.js - o librerie Javascript ce facilitează integrarea soluției în sistemele actuale.

Pe lângă acestea sunt prezente și două situri web de tip documentație și un sit web de prezentare ce au ca scop oferirea utilizatorilor o imaginea clara asupra soluției propuse și a modului în care aceasta funcționează.

## 1.3 Cerințe

În cadrul acestui subcapitol se vor prezenta cerințele sistemului propus. Prin prisma funcționalității, cerințele sunt de două tipuri: funcționale și non-funcționale. Cerințele funcționale definesc funcțiile unui sistem și a componentelor acestuia, iar cele non-funcționale surprind criteriile care pot fi folosite pentru a analiza aspectele legate de operationalitatea sistemului.

## Cerințe funcționale

### **1. Crearea unei identități digitale ale tuturor participanților dintr-un lanț de aprovisionare**

Companiile de producție sau transport, precum și cele de certificare vor trebui să dețină o identitate digitală pentru a interacționa cu restul funcționalităților.

### **2. Crearea unei identități digitale ale materialelor și ale loturilor**

Fiecare material și lot de materiale vor trebui să corespundă cu un echivalent în mediul virtual, pentru a facilita trasabilitatea acestora.

### **3. Abilitatea de a înregistra fiecare transport a loturilor**

Transporturile și schimburile de proprietate ale materialelor reprezintă un eveniment important în istoricul acestora, iar înregistrarea acestor evenimente este esențială.

### **4. Abilitatea de a asigna certificate către companii sau materiale**

Asignarea de certificate către companii și materiale este benefică atât pentru companii cât și pentru consumatori, deoarece acestea transmit informații cu privire la calitatea produselor sau a practicilor de fabricație ale acestora.

### **5. Posibilitatea consumatorilor de a vedea istoricul materialelor**

Consumatorii vor putea vedea informații cu privire la istoricul produselor printr-o interfață web ușor accesibilă.

## Cerinte non-funcționale

### **1. Securitate**

Integritatea și disponibilitatea datelor sunt esențiale atât pentru participanții lanțurilor de aprovisionare cât și pentru consumatori.

### **2. Scalabilitate**

Sistemul trebuie proiectat astfel încât modelarea proceselor, precum crearea de materiale și loturi noi, să fie cat mai generală, pentru a putea avea o aplicabilitate cat mai largă.

### **3. Usurința integrării**

Deoarece la momentul actual multe din companii utilizează un program de administrare a inventarului sau a lanțului de aprovisionare, sistemul de trasabilitate vine ca o extensie a programelor existente. Acesta trebuie să fie ușor integrabil.

## 1.4 Funcționalitățile aplicației

### 1.4.1 Introducere

Sistemul prezintă 4 tipuri de actori, companii, autorități de certificare consumatori și administratorul de sistem.

**Companiile** sunt de 3 tipuri:

- **Fabrici** - companiile din acest tip pot crea diferite materiale, pot crea loturi de materiale și pot iniția transporturi către alte companii.
- **Depozite** - pot primi loturi de la alte companii și iniția transporturi
- **Retail** - Similar cu tipul de companie depozit
- **Transport** - companiile de tip transport intermediaza transportul loturilor de materiale de la o companie la alta.

**Autoritățile de certificare** pot administra certificatele, precum crearea, asignarea și revocarea acestora.

**Consumatorii** pot vedea istoricul produsului prin introducerea unui număr unic sau scanarea unui cod QR.

**Administratorul de sistem** este reprezentat în mod digital prin adresa ce a încărcat contractele inteligente pe blockchain. Acesta poate revoca certificatele asignate materialelor sau companiilor și modifica suma minima de plată pentru atribuirea unui certificat.

Figura 4 prezintă diagrama use-case a aplicației.

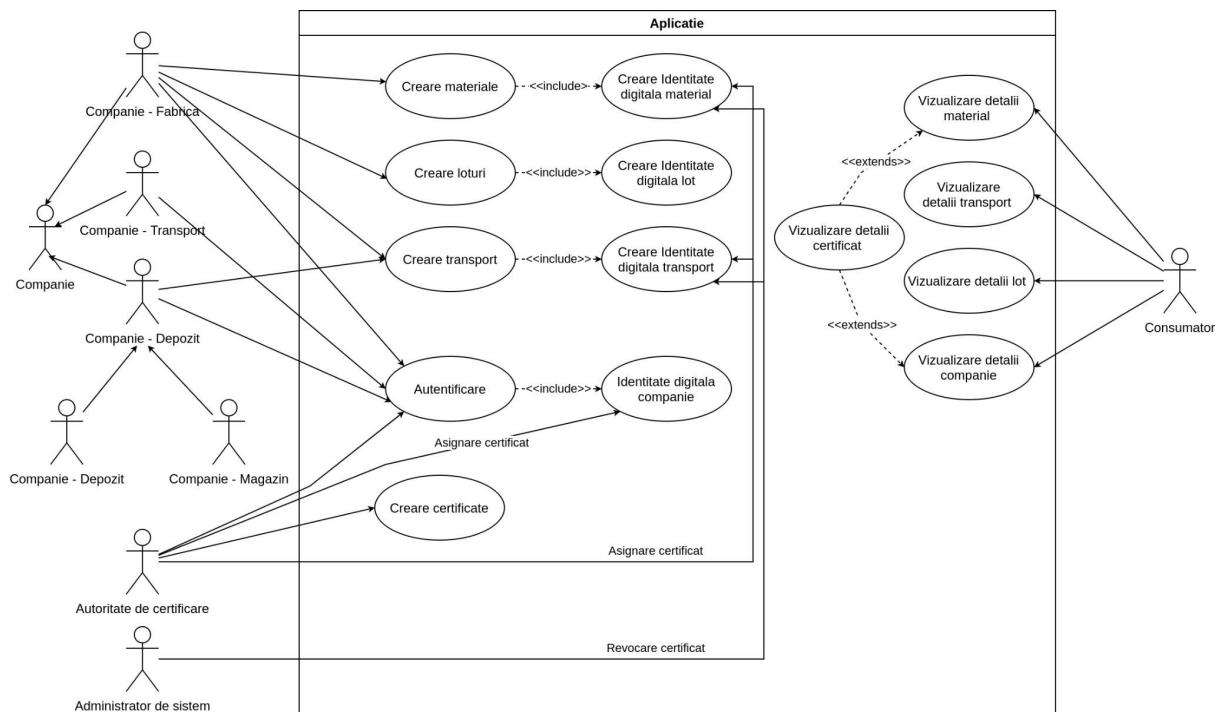


Figura 4: Diagrama use case

## Entități

Solutia prezinta mai multe tipuri de entitati precum: materiale, loturi, transporturi, certificate.

## Materiale

Pentru a dezvolta un sistem de trasabilitate cat mai precis, fiecare material trebuie sa dețină un echivalent în mediul digital. Pentru a realiza acest lucru, am folosit o structura de tipul *bill of materials (BOM)* [43]. Aceasta reprezintă un mod ierarhic de reprezentare a materialelor și a componentelor din care sunt făcute.

Materialele sunt clasificate în două categorii:

- Materialele de baza (Raw Materials)
- Materialele compuse (Materials) - sunt acele materiale care, pentru a fi create, avem nevoie de alte materiale, conform unei rețete specificate în momentul definirii acestuia. Figura 5 prezinta o definiție a unui material compus sub forma unei structuri arborescente.

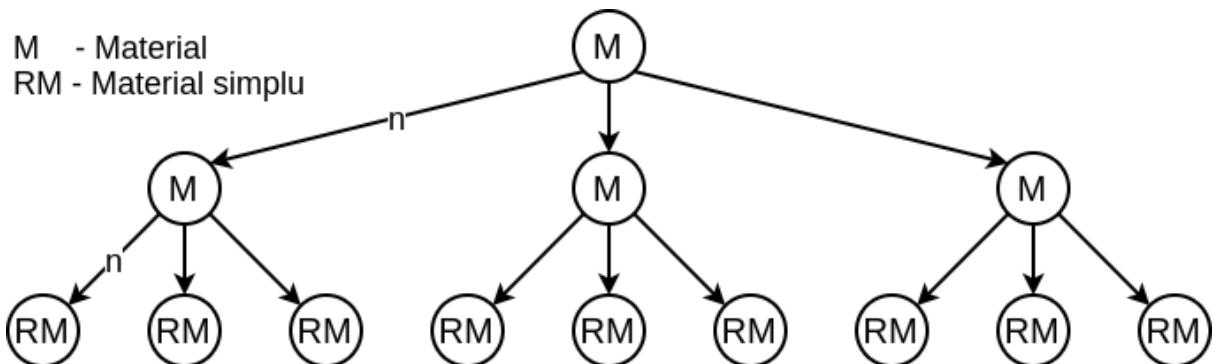


Figura 5: O structura abstractă a unui material

Fiecare companie ce își dorește să creeze un produs, trebuie de asemenea să creeze o definiție a acestuia. Pentru a defini un material, avem nevoie de următoarele informații:

- Numele materialului
- Un identificator pentru o unitate a materialului (ex: kg, m, “300g”, item)
- (optional) O tupla formată din 2 element (ID, cantitate)
  - ID: ID-ul materialului necesar pentru crearea a o unitate
  - Cantitate: numărul de unități a materialului specificat de ID

După cum putem vedea în exemplul din Figura 6, nodurile frunza sunt reprezentate de materiale simple, ce nu conțin o rețetă, iar restul sunt noduri asociate materialelor compuse din mai multe ingrediente.

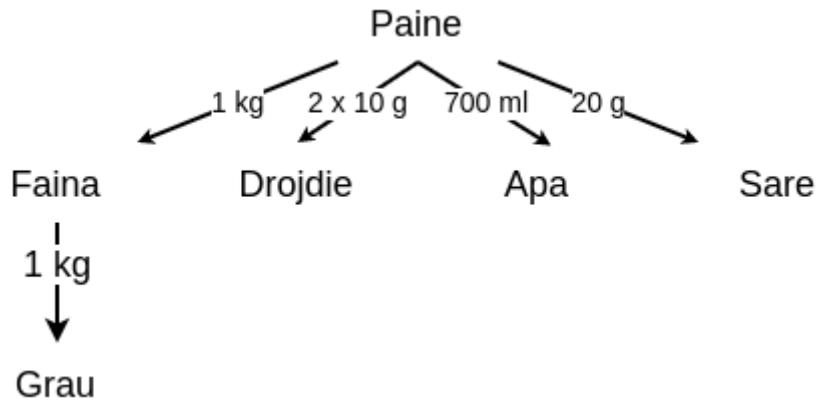


Figura 6: Un exemplu al definiției arborescente a ingredientelor unui material

Pentru a crea o instanță a unui material simplu, nu avem nevoie de niciun fel de date de intrare, iar pentru o instanță a unui material compus, avem nevoie de loturile cu produsele asociate rețetei acestora. Specificarea loturilor se face printr-un vector de tuple de tipul [...] (ID lot, [...ID Material]), specificandu-se astfel materialele de intrare și loturile în care acestea se află. În Figura 7, este prezentat grafic modul de crearea a unui material compus. Ca intrare sunt specificate loturile împreună cu materialele din interiorul acestora ce vor fi folosite, iar apoi, pe baza ingredientelor necesare ale materialului compus, se calculează cele rămase.

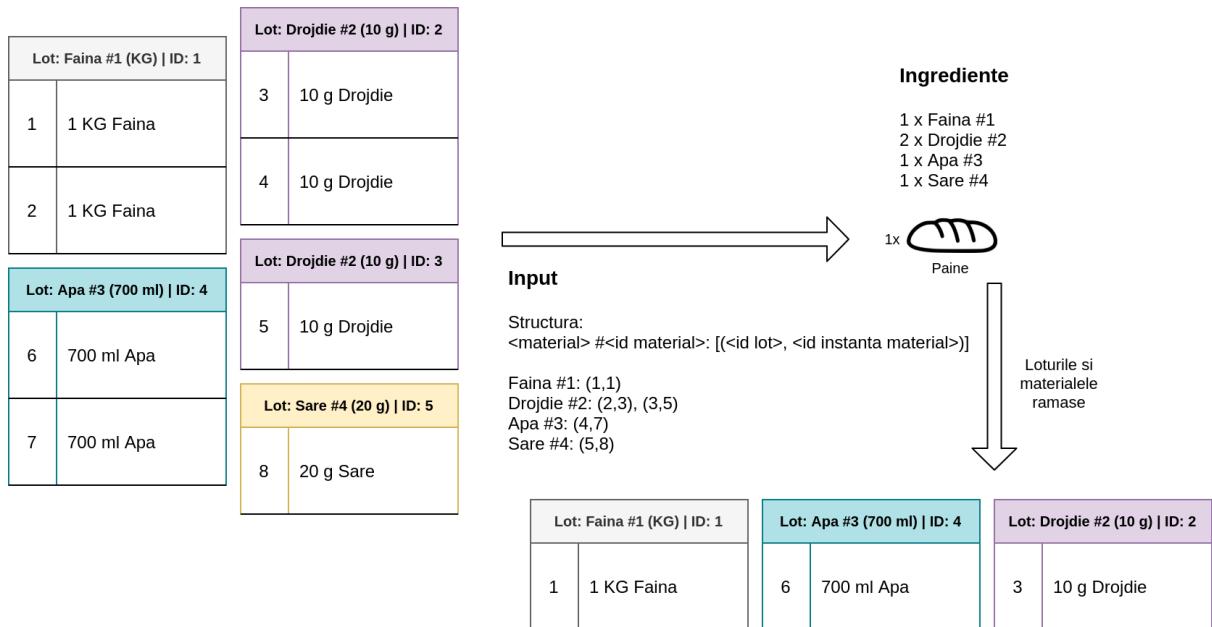


Figura 7: Un exemplu de creare al unui material compus

## Loturi

Loturile sunt o mulțime de instanțe de materiale de același tip.

## Transporturi

Transporturile reprezintă un schimb al proprietarilor unor loturi de produse. Companiile ce detin loturile pot iniția transporturi către alte companii, specificând următoarele informații:

- Destinatarul - adresa Ethereum a companiei ce urmează să primească loturile
- Compania de transport - adresa ethereum a companiei ce va transporta loturile
- (optional) O parolă - pentru a schimba deținătorul loturilor la nivelul contractelor inteligente, destinatarul trebuie să introducă parola setată de inițiatorul transportului.  
În acest mod se poate reduce fraudă în cazul în care lotul nu ajunge la destinația corectă.

După inițierea transportului, compania ce efectuează transportul poate seta anumite stări, precum:

- **Ready for Transit (READY\_FOR\_TRANSIT)** - Transportul este pregătit de încărcare pentru a putea fi trimis către destinație
- **Pending Transit (PENDING\_TRANSIT)** - Transportul este încărcat și așteaptă să fie trimis către destinație
- **In Transit (IN\_TRANSIT)** - Transportul este în transit
- **Pending Finalised (PENDING\_FINALISED)** - Transportul a ajuns la destinație și așteaptă să fie finalizat de către destinatar

După setarea ultimei stări, pentru ca la nivel contractual să fie schimbat proprietarul loturilor, destinatarul trebuie să seteze starea transportului astfel:

- **Finalised (FINALISED)** - marchează un transport finalizat cu succes. În cazul în care acesta are specificat o parolă, aceasta trebuie transmisă. Dacă parola este incorrectă sau nu este specificată, transportul nu poate primi starea și implicit schimbul de proprietate asupra loturilor nu este realizat.

## Certificate

Certificatul reprezintă o afirmație cu privire la calitățile și impactul unui produs sau a unei companii. Acestea sunt create și atribuite de către autoritatele de certificare. În momentul atribuirii, autoritatea de certificare trebuie să plătească o sumă minimă, sumă ce este stabilită initial în contractele inteligente, dar care poate fi modificată de către administratorul de sistem ulterior. În cazul în care, compania sau produsul căruia i-a fost asignat certificatul nu mai corespunde cu afirmația acestuia, există două variante prin care poate fi revocat:

- Compania de certificare revoca certificatul, iar suma plătită în momentul asignării este returnată
- Administratorul de sistem revoca certificatul, dar suma plătită nu este returnată către compania de certificare.

Acest concept este inspirat din algoritmul Proof Of Stake, deoarece prin acest mod se poate descuraja asignarea de certificate către produse și companii care nu corespund, iar companiile de certificare sunt încurajate să verifice periodic dacă afirmația certificatului este respectată.

Certificatele pot fi de mai multe tipuri, în funcție de mesajul transmis de acestea, precum:

- certificate care descriu impactul asupra mediului pe care îl are un material sau o companie. Un exemplu pentru descrierea acestui tip poate fi: "Resursele utilizate pentru fabricarea acestui produs sunt gestionate într-un mod care să asigure sănătatea și întreținerea pe termen lung a acestor resurse."
- certificate care descriu siguranța utilizării unui material sau al procesului de producție al acestuia. Un exemplu pentru descrierea acestui tip poate fi: "Produsul a fost testat pe oameni într-un studiu clinic pentru a asigura siguranța."
- certificate care descriu beneficiile ingredientelor produselor sau modul în care acestea au fost fabricate. Un exemplu pentru descrierea acestui tip poate fi: "Produsul nu conține arome sintetice. Poate conține arome derivate din ingrediente naturale."
- certificate care descriu impactul social pe care îl are cumpărarea de produse sau practicile companiei. Un exemplu pentru descrierea acestui tip poate fi: "Compania are politici de egalitate și diversitate în vigoare pentru a se asigura că accesul la drepturi sau oportunități nu este afectat."
- certificate care descriu impactul pe care îl are produsul asupra animalelor sau practicile de creștere ale acestora. Un exemplu pentru descrierea acestui tip poate fi: "Produsul a fost dezvoltat și fabricat într-un mod care nu dăunează sau ucide animalele."
- un tip generic, în cazul în care certificatul nu se încadrează în cele de mai sus

#### 1.4.2 Autentificarea

Autentificarea este realizată prin 3 moduri. Metamask, pentru autentificarea prin extensia Metamask, Torus pentru autentificarea prin Google, și prin fraza mnemonica ce este echivalentul cheii private. Pagina de autentificare conține butoane pentru a selecta modul dorit, precum este prezentat în Figura 8.

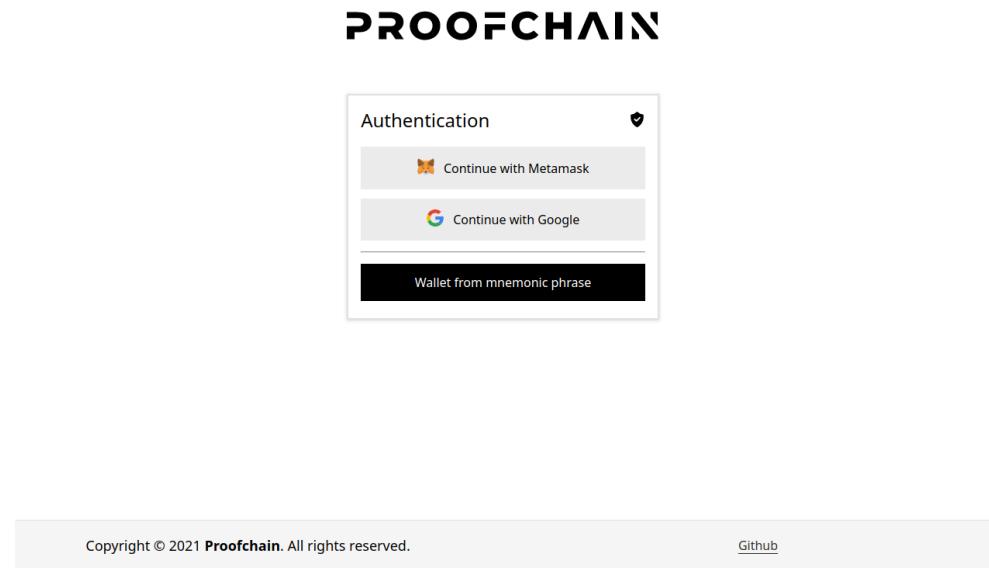


Figura 8: Alegerea modurilor de autentificare

## Metamask

Metamask este un wallet Ethereum sub forma unei extensii a browser-ului. Pe langa functionalitatile specifice unui wallet, aceasta dispune și de funcționalitatea de a conecta aplicațiile web la blockchain. Figura 9 evidențiază momentul în care trebuie să specificam un cont (pereche cheie publică/privată) pe care îl vom folosi pe parcursul aplicației.

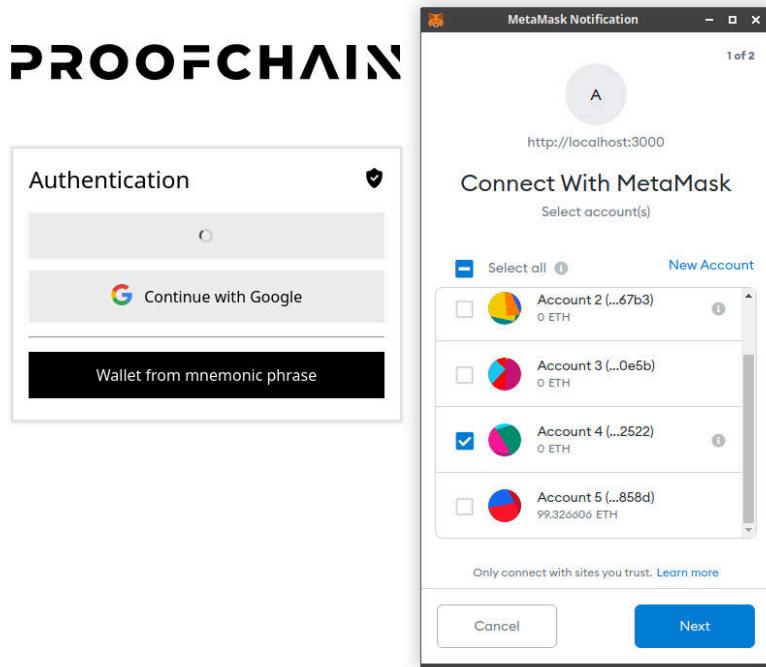


Figura 9: Selectarea unui cont în autentificarea cu Metamask

## Torus

Torus [41] este o soluție de autentificare ce combina tehnologia blockchain cu diferite metode de autentificare precum OAuth, OAuth2, AWS Cognito etc., pentru a pune la

dispoziția aplicațiilor descentralizate un mod non-custodial de a genera chei private și autentifica utilizatorii.

În cadrul aplicației, Torus este folosit pentru a facilita autentificarea cu contul Google. Figura 10 prezintă fereastra specifică autentificării cu contul Google.

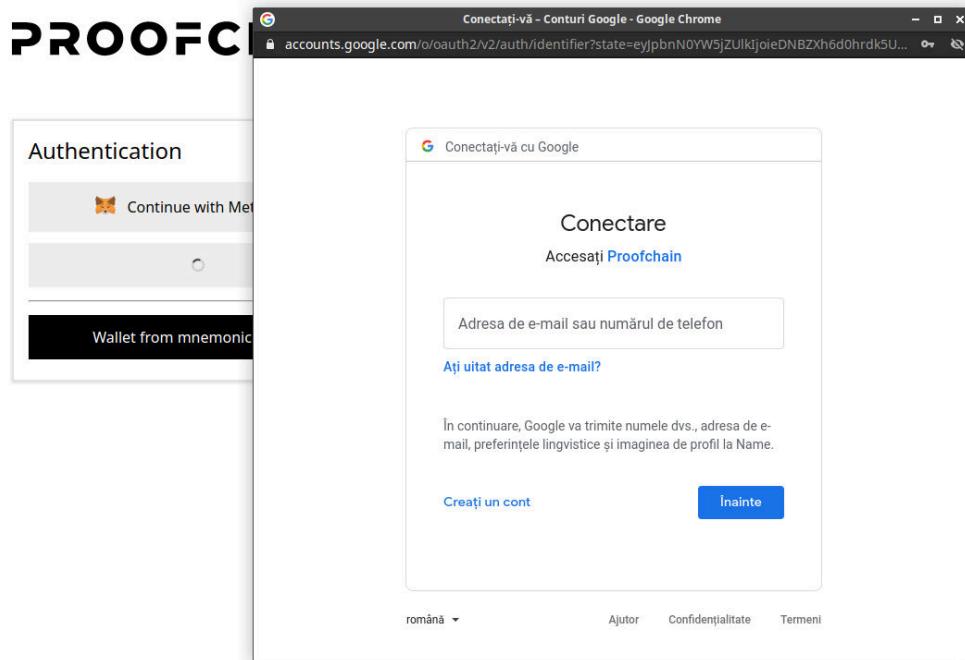


Figura 10: Autentificarea cu contul Google

### Fraza mnemonică

O frază mnemonică este o listă de 12 cuvinte care acționează ca un generator pentru o cheie privată. Aceasta este folosită pentru a ușura memorarea de către oameni a cheii private. În Figura 11 este prezentată fereastra specifică autentificării prin această modalitate.

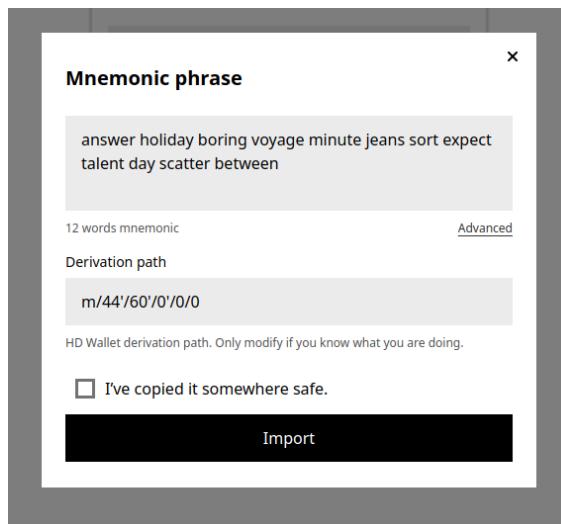


Figura 11: Autentificarea cu ajutorul unei fraze mnemonice de 12 cuvinte

#### 1.4.3 Crearea unei identități

În cazul în care contul este pentru prima dată creat și nu are o entitate asociată în blockchain, utilizatorului îi este prezentată o pagina ce permite crearea unei identități. În funcție de tipul de identitate ales, formularul își schimba câmpurile aşa cum este evidențiat în Figura 12.

The figure consists of three screenshots of the PROOFCHAIN application interface, illustrating the steps for creating a new entity:

- Screenshot 1 (Left):** Shows the "Initial setup" screen with the heading "Initial setup" and the sub-instruction "Your account doesn't have any associated entities". It displays two steps:
  - Step 1:** "Select entity type" with two options:
    - Company**: Described as "Creates, transports and manufactures materials". This option is selected.
    - Certificate Authority**: Described as "Emits certificates for companies and materials".
  - Step 2:** "Fill information".At the bottom are "Previous" and "Next" buttons.
- Screenshot 2 (Top Right):** Shows the "Initial setup" screen after selecting "Company". The "Select entity type" step is marked with a checkmark. The "Fill information" step is active, showing fields for "Entity Name" (with placeholder "Entity Name") and "Entity type" (set to "Manufacturer"). At the bottom are "Previous" and "Create entity" buttons, where "Create entity" is highlighted.
- Screenshot 3 (Bottom Right):** Shows the "Initial setup" screen after filling in the "Entity Name" field. The "Entity Name" field now contains "Entity Name". The "Create entity" button is still highlighted.

Figura 12: Formularul de creare al unei identități

#### 1.4.4 Administrarea materialelor

După autentificarea cu success a unui utilizator ce deține deja o identitate digitală sau după crearea acesteia, interfața din Figura 13 este afișată. Aceasta conține informații generale cu privire la profilul și activitatea utilizatorului. De asemenea, se poate observa structura aplicației: în partea stanga este un meniu de navigare laterală ce conține link-uri pentru navigarea către alte pagini, iar în partea de sus este un meniu de navigatie ce conține adresa Ethereum a companiei, balanta și un meniu cu numele acestuia.

Figura 13: Pagina principală

Panoul de administrare prezintă două secțiuni pentru materiale, fiecare având o pagină pentru vizualizare și una pentru adăugare:

- “Materials” - Secțiune ce este destinată materialelor compuse
- “Raw Materials” - Secțiune ce este destinată materialelor simple

În Figura 14 sunt evidențiate paginile specifice listării și creării unor materiale simple (“Raw Materials”). Crearea unui material simplu constă în completarea unui formular cu numele materialului, un cod arbitrar și un identificator al unei instanțe al materialului.

Figura 14: Listarea și crearea unui material simplu

În contextul materialelor compuse (“Materials”), pagina de listare ale acestora este asemănătoare cu cea a materialelor simple. O diferență importantă este în modul de creare ale acestora, formularul continând și o secțiune dinamica de specificare a ingredientelor precum în Figura 15.

De asemenea, acțiunea de creare a unui material implica o tranzacție la nivelul platformei Ethereum, iar la fiecare tranzacție utilizatorului îi este afișat un dialog, precum în Figura 16, de confirmare ce conține informații cu privire la costurile asociate tranzacției și un mod de a ajusta valoarea unității de gas, în cazul în care acesta dorește.

Create Material

Material name

Descriptive material name

Material code

Optional material identification code

Amount identifier

Kilogram - kg

Material Token Id

Amount

1

+ Add material

Create material

Figura 15: Formularul de creare al unui material compus

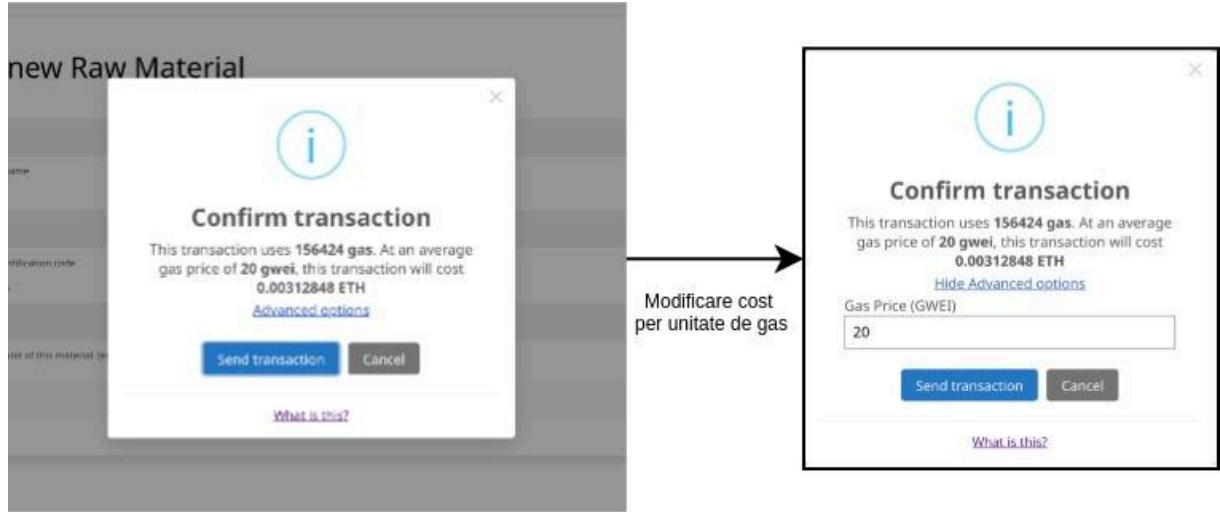


Figura 16: Dialogul de confirmare și modificare al costului unei tranzacții

Pagina de vizualizare conține informații cu privire la caracteristicile produsului precum: nume, ingrediente (în cazul în care produsul este unul compus) și instanțele acestuia deținute de contul curent, dar care nu sunt asignate unui lot, precum este evidențiat în Figura 17. În partea dreapta, este prezent tabelul de instanțe ale materialului, împreună cu uuid-ul acestora și hash-ul tranzacției de creare. Apasarea pe uuid-ul instanței afisează o fereastră cu

un cod QR și cu posibilitatea de a descărca o imagine png cu respectivul cod, ca în Figura 18.

The screenshot shows the ProofChain application interface. On the left, there's a sidebar with navigation links: Dashboard, Materials (All materials, Create material), Raw materials, Batches, Transports, Certificates, GitHub, and Documentation. The main content area is titled 'material > 6'. It shows a table for the 'Bread' material with the following data:

Material Token Id	Material name	Amount
0	Wheat Grain	1 kg
1	Yeast	2 gram
2	Salt	5 gram
3	Sugar	3 kg

Below this is a section titled 'Balance - 1' with a table:

Current balance: 1	Materials uid
Batch Id: 0	3 x 2 x 0

A note says: 'The batch id that contains materials Materials uid from the batch that will be used to mint this material.'

There are also sections for 'Batch Id' and 'Materials uid' with dropdown menus. A button '+ Add batch' is present. Below these is a large black button labeled 'Mint one "100 grams" of Bread'.

To the right, under 'Your materials', there's a table:

UUID	Material ID	Mint Transaction
61	6	0x21e89a74...ade053305

Figura 17: Informații despre un material compus.

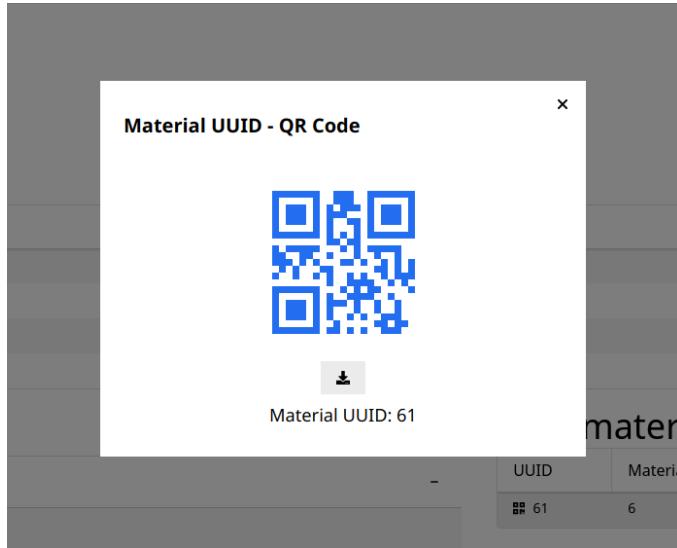


Figura 18: Codul QR ce reprezinta uuid-ul produsului

În partea stângă, pagina de vizualizare al unui produs conține și un panou format din două secțiuni, prima este cea de creare al unei noi instanțe, iar a doua afișează toate certificatele asignate materialului. În funcție de tipul de material, formularele de creare al unei instanțe sunt diferite, precum este ilustrat și în Figura 19. În cazul în care acesta este compus, un formular dinamic este afișat pentru a selecta loturile și materialele din interiorul loturilor

ce vor fi folosite, iar pentru materialele simple este necesara specificarea doar a numarului de unitati.

Figura 19: Formularele de creare al unei instanțe de material

Certificatele asociate materialului selectat sunt afișate în a două secțiune a panoului, sub forma unui tabel. Detaliile specifice fiecărui certificat pot fi văzute prin apăsarea unui buton, precum în Figura 20.

Name	Certificate Authority	Stake	Time	Action
Protect animals	Certificate Company	2 ETH	4 hours ago	<a href="#">View</a>
Natural packaging	Certificate Company	2 ETH	4 hours ago	<a href="#">View</a>

Assignment details		Certificate Info	
Name	Protect animals	Description	The product was developed and manufactured in a way that does not harm or kill animals.
Code	0	Certificate Authority Address	0x7944cE684e4b57b8901b2cD81B17809e3b229035
Created	4 hours ago	Created Transaction	0x0c68d322...adc3ae315

Assignment details		Certificate Info	
Name	Certificate Company	Description	The product was developed and manufactured in a way that does not harm or kill animals.
Code	0	Certificate Authority Address	0x7944cE684e4b57b8901b2cD81B17809e3b229035
Created	4 hours ago	Created Transaction	0x41ad3f97...161cb0442

Figura 20: Lista certificatelor și detaliile unui certificat.

## 1.4.5 Administrarea loturilor

În Figura 21 este evidențiată pagina de afișare ale loturilor. Aceasta conține un tabel cu toate loturile împreună cu id-ul fiecărui, codul, id-ul materialului, uuid-urile instanțelor materialului și hash-ul tranzacției de creare.

ID	Code	Material Id	Material Uuids	Created Transaction	Action
17	LOT_N_9542201	5	(58) (59) (60)	0xabe1082c...4f8b833e5	<button>View</button>
16	LOT_N_8999558	5	(54) (55) (56) (57)	0xdd42ff1f4...c6a04df5e	<button>View</button>
15	LOT_N_4032997	5	(51) (52) (53)	0xd6e4862b...76cf021e1	<button>View</button>
14	LOT_N_2379901	4	(48) (49) (50)	0x371c6bd9...c2ffa115b	<button>View</button>
13	LOT_N_296356	4	(44) (45) (46) (47)	0xe51a254c...7c3392b8f	<button>View</button>
12	LOT_N_5997941	4	(41) (42) (43)	0xd9faba45...aa2326f52	<button>View</button>
5	LOT_N_1517902	1	(18) (19) (20)	0x220e97b9...458de1440	<button>View</button>
4	LOT_N_5460420	1	(14) (15) (16) (17)	0xdf8de1ba...d5079dfe3	<button>View</button>
3	LOT_N_4275011	1	(13)	0xef735e18...213bba72b	<button>View</button>
2	LOT_N_4067251	0	(8) (9) (10)	0x35d0da45...1c5ac9c12	<button>View</button>

Figura 21: Lista de loturi

Precum este evidențiat în Figura 22, crearea unui lot este realizată cu un formular ce va conține uuid-urile instantelor materialelor ce vor fi adăugate în lot și codul acestuia.

Figura 22: Creearea unui lot nou

Vizualizarea unui lot este evidențiată în Figura 23. Aceasta pagina conține în partea stanga detalii cu privire la metadatele lotului (id și cod) și hash-ul tranzacției de creare, iar în partea

dreapta instanțele ale materialului conținut de lot. De asemenea, prin apăsarea butonului din dreapta sus, “Destroy Batch”, lotul este șters, iar materialele conținute vor apărea din nou în tabelul cu toate materialele și vor putea fi adăugate din nou într-un alt lot.

UUID	Material ID	Mint Transaction
# 58	5	<a href="#">0xee169f59...5bd47ee93</a>
# 59	5	<a href="#">0xee169f59...5bd47ee93</a>
# 60	5	<a href="#">0xee169f59...5bd47ee93</a>

Figura 23: Detalii lot

#### 1.4.6 Administrarea transporturilor

Pagina ce prezinta toate transporturile este prezentata în Figura 24. Aceasta conține un tabel cu informații generale ale fiecărui transport precum adresa Ethereum a destinatarului, adresa Ethereum a companiei de transport, id-urile loturilor, starea și hash-ul tranzacției de creare al lotului.

Id	Receiver	Transport Company	Batch ids	Status	Created Transaction	Action
5	0x92f28A28...C2FC3D0D3	0x15Af53F1...0A90197B9	12, 13	No status	<a href="#">0x65032e55...24e9e68e9</a>	<a href="#">View</a>
4	0x92f28A28...C2FC3D0D3	0x15Af53F1...0A90197B9	3, 16	Ready for transit	<a href="#">0xd1d9636e...d75271b6b</a>	<a href="#">View</a>
3	0xD2f19b14...F2660c858	0x15Af53F1...0A90197B9	11	Finalised	<a href="#">0xe05b1770...63b9fa6d9</a>	<a href="#">View</a>
2	0xD2f19b14...F2660c858	0x15Af53F1...0A90197B9	9, 10	Finalised	<a href="#">0xb8242bc9...3f7282860</a>	<a href="#">View</a>
1	0xD2f19b14...F2660c858	0x15Af53F1...0A90197B9	8	Finalised	<a href="#">0xb59ae45d...035d34edb</a>	<a href="#">View</a>
0	0xD2f19b14...F2660c858	0x15Af53F1...0A90197B9	6, 7	Finalised	<a href="#">0x3f8ba46f...eb3100101</a>	<a href="#">View</a>

Figura 24: Tabel cu transporturi

In Figura 25 se poate observa pagina de vizualizare a unui transport finalizat. În partea centrală este afișat un istoric al stării acestuia, în dreapta sunt afișate informații cu privire la loturile transportate, iar în stânga sunt alte detalii relevante.

The screenshot shows the PROOFCHAIN platform interface for a completed transport. On the left, a sidebar lists navigation options: Dashboard, Materials, Raw materials, Batches, Transports (selected), All transports, Create transport, and Certificates. Below the sidebar, there are links to GitHub and Documentation.

The main content area has a header "Transport information" with a back button. It displays a timeline of events:

- Finalised 9 hours ago: Transport arrived at the destination (Details)
- Pending finalisation 9 hours ago: Transport is waiting receiver confirmation (Details)
- In transit 9 hours ago: Transport is in transit (Details)
- Pending transit 9 hours ago: Transport is waiting to be loaded (Details)
- Ready for transit 9 hours ago: Transport is ready to be sent to the receiver (Details)
- Transport initiated 9 hours ago: Transport entry created (Details)

Below the events, there are two tables:

Transport details		Transport Contents		
ID	3	Batch Id	Batch Code	Material Id
Current status	Finalised	11	LOT_N_7792903	3
Password Protected	Yes			(38) (39) (40)
Created	9 hours ago			
Created Transaction	<a href="#">0xe05b1770...63b9fa6d9</a>			

At the bottom, there is a link "View transaction details".

Figura 25: Vizualizare detalii transport

Schimbarea stării unui transport este realizată inițial de compania de transport specificată la crearea acestuia. În Figura 26 este prezentată pagina de vizualizare din contul unei companii de transport. Spre deosebire de Figura 25, aceasta prezintă și un formular de schimbare al stării.

În Figura 27 este evidențiat modul și ordinea în care sunt schimbrate stările unui transport. Expeditorul îl creează, compania de transport modifică anumite stări asociate evenimentelor fizice, iar destinatarul confirma finalizarea acestuia.

The screenshot shows the 'Transport information' page for a transport entry. The top navigation bar includes 'Dashboard', 'Transports' (with 'All transports' and 'Certificates' sub-options), and a back arrow. The main content area has a breadcrumb path: 'transport > 5'. The title 'Transport information' is centered above a section for 'Events'. An event log shows 'Transport initiated' (21 hours ago) and 'Transport entry created'. A 'Status' dropdown is set to 'Ready for transit', with a note: 'Select the status you want to assign to this transport'. A 'Set status' button is present. Below this is a 'Transport details' table with columns for Id (5), Current status (No status), Password Protected (Yes), Created (21 hours ago), and Created Transaction (a long hex string). To the right is a 'Transport Contents' table with columns for Batch Id, Batch Code, Material Id, and Material Uuids, listing two entries: LOT\_N\_5997941 and LOT\_N\_296356, each associated with four material UUIDs (41, 42, 43, 44, 45, 46, 47). At the bottom left are links for 'Github' and 'Documentation'.

Figura 26: Pagina unui transport din perspectiva unei companii de transport

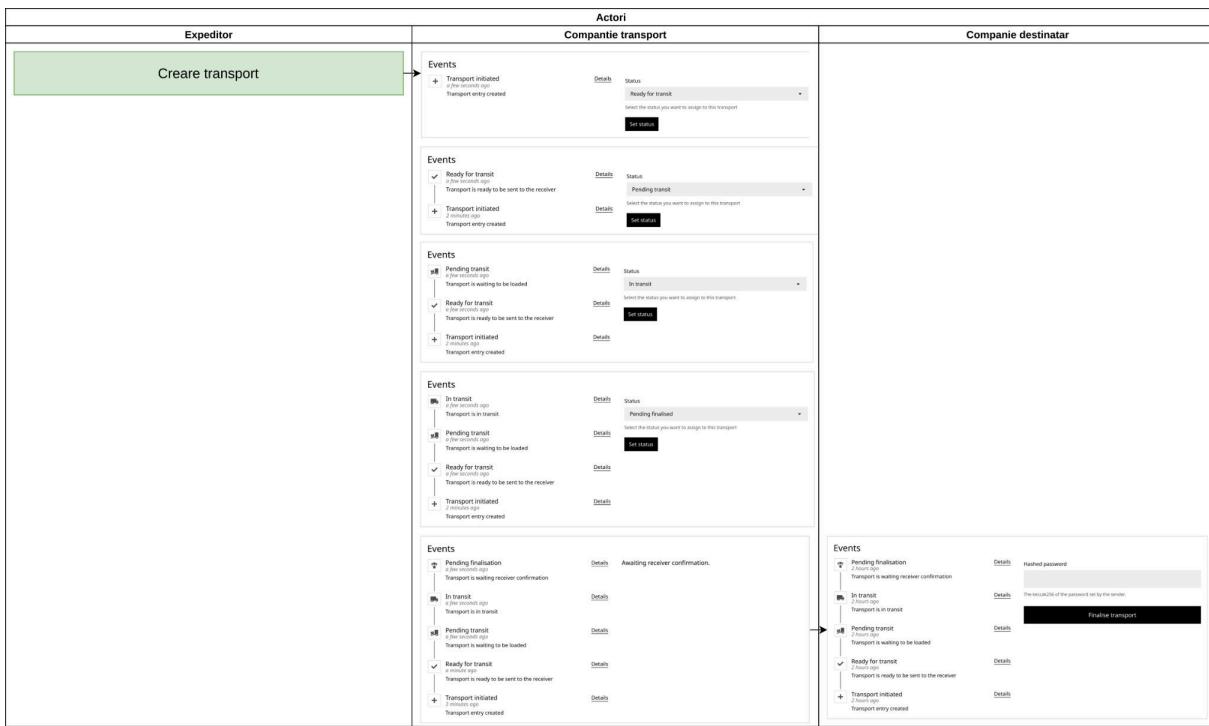


Figura 27: Ordinea executarii operatiilor asupra unui transport

#### 1.4.7 Administrarea certificatelor

Figura 28 prezintă pagina cu lista certificatelor asignate unei companii.

Figura 28: Certificatele atribuite companiei

Apasand butonul de vizualizare, în Figura 29, putem vedea informații cu privire la detaliile certificatului (dreapta) și istoricul asignării certificatului (stanga).

Figura 29: Vizualizare certificat asignat unei companii

Crearea unui certificat este realizată de autoritatea de certificare printr-un formular, specificând numele, o scurta descriere și tipul de certificat. În Figura 30 putem vedea atât formularul de creare cât și formularele de asignare către o companie sau un material.

În cazul asignării către o companie, autoritatea de certificare trebuie să specifică codul certificatului, adresa companiei și garanția, iar în cazul asignării către un material, în loc de adresa companiei va fi specificat id-ul materialului. De asemenea fiecare camp din formulare are validare, pentru a se asigura că adresa companiei, id-ul materialului sau codul certificatului există pe blockchain.

The figure consists of three side-by-side screenshots of the Proofchain web application. The left screenshot shows the 'Create certificate' page with fields for 'Certificate name', 'Description', and 'Type' (set to 'Animal Welfare'). The middle screenshot shows the 'Assign certificate' page for a company, with fields for 'Company' and 'Material'. The right screenshot shows the same 'Assign certificate' page but for a material, with fields for 'Material' and 'Material Id'. All pages include a 'Create Certificate' or 'Assign Certificate' button at the bottom.

Figura 30: Paginile de creare și asignare al unui certificat

După creare, certificatele pot fi văzute într-un tabel similar cu cel al materialelor sau mai specific, intr-o pagina ce prezinta informatii cu privire la un anumit certificat și la materialele sau companiile la care acesta este asignat, precum în Figura 31. În tabelul din partea stanga, sau in cel din dreapta în cazul în care certificatul este asignat unui material, este prezent și un buton ce anuleaza asignarea certificatului.

The screenshot shows a detailed view of a certificate titled 'Reduced Material Waste'. It includes fields for 'Name' (Reduced Material Waste), 'Description' (The business has implemented measures to reduce pre and post consumer product waste.), 'Type' (Environmental Impact), and 'Certificate authority address' (0x7944cE6B4e4b57b8901b2cD81B178D9e3b229035). Below this, a section titled 'Assigned to:' shows a table of materials assigned to the certificate. To the right, another table lists companies that have accepted the certificate, each with a 'Cancel' button next to it. The tables have columns for Company name, Company type, Stake, Time, Transaction, and Actions.

Figura 31: Vizualizare certificat din perspectiva unei autorități de certificare

#### 1.4.8 Vizualizarea istoricului unui produs

Aplicația client permite consumatorilor să vizualizeze detalii despre anumite produse, acest lucru fiind bazat pe introducerea sau scanarea unui cod unic asociat fiecărui produs. Din punct de vedere tehnic, aceasta este o extensie al panoului de administrare, codul fiind în același proiect. Inițializarea librăriei ce folosește Web3.js este realizată similar, dar, deoarece

clientul nu poate efectua tranzacții pe blockchain, nu se mai specifică o pereche de cheie publică și privată.

### Selectare material

Fiecare material este identificat într-un mod unic de un uuid generat la instanțierea acestuia din panoul de administrare al companiei. Pentru usurința folosirii aplicației atât pe dispozitivele mobile cât și pe cele de tip desktop, există două moduri de selectare al unui material, aşa cum este prezentat și în Figura 32: scanare al unui cod QR sau introducerea manuală al acestuia.

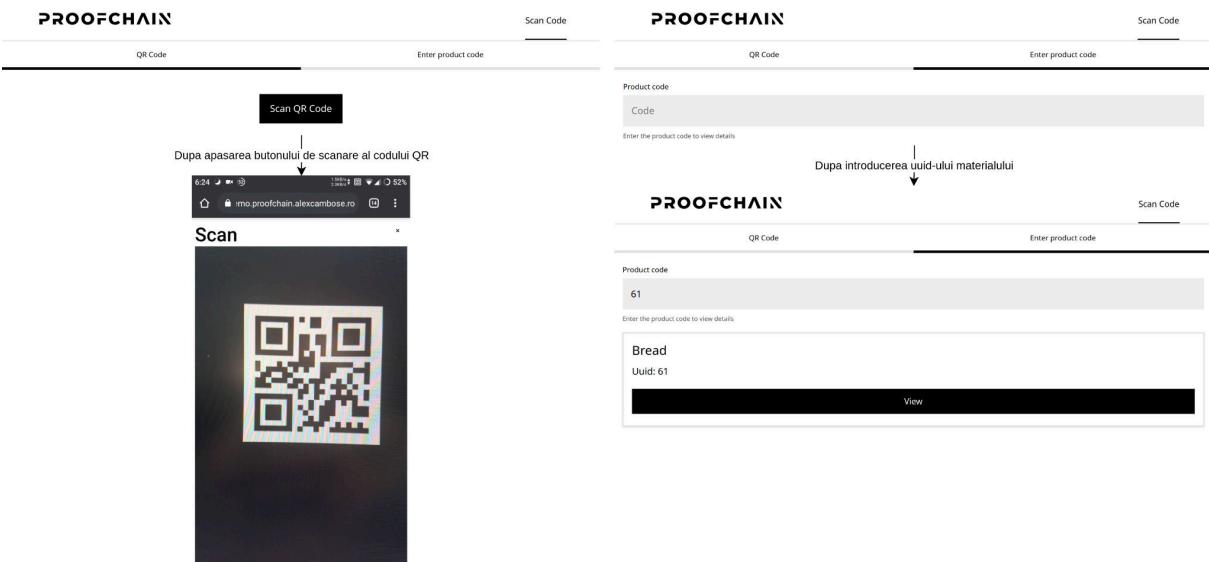


Figura 32: Pagina de scanare sau introducere al codului

### Vizualizare detalii

După selectarea materialului prin scanarea unui cod QR sau prin introducerea acestuia, aplicația client redirectionează utilizatorul către o pagină similară cu cea din Figura 33. Aceasta conține mai multe secțiuni “Information”, “Material graph” și “Company”, prima fiind bazată pe detalii cu privire la caracteristicile materialului, istoricul acestuia și certificatele asignate.

Selectând opțiunea “Material graph”, în Figura 34, putem observa o structură arborescentă al unui material compus. Elementul rădăcina, colorat violet, este cel selectat de utilizator pentru a-i se vedea detalii. Acesta este compus din mai multe materiale, nodurile albastre, adăugate în loturi, nodurile galbene. Muchiile conțin hash-ul asociat tranzacției de creare al lotului, în cazul în care aceasta conectează un nod de tip material (albastru) cu un nod de tip lot (galben). De asemenea, muchia va conține hash-ul tranzacției asociate creării unui material nou, în cazul în care aceasta conectează un nod de tip slot cu unul de tip material. Arborele este realizat prin interogarea evenimentelor emise de contractele inteligente (Ethereum event logs).

The figure consists of three screenshots of the PROOFCHAIN platform interface, each showing different levels of detail for a material named "Bread".

- Screenshot 1:** Shows the "Information" tab for material ID Ju73. It displays fields such as Name (Bread), Code (Ju73), Amount Identifier (100 grams), Creator (0x201161-42d45eAFD0a2D2457d7e5eF2660c858d), and Create Event (0x21e93a74\_ae053305). A "Scan Code" button is at the top right.
- Screenshot 2:** Shows the "Material graph" tab for the same material. It displays a hierarchical tree where "Bread" is the root node, which branches into "LOT N\_7407366", "LOT N\_3050842", "LOT N\_2155840", "LOT N\_B026204", and "LOT N\_1091615". Each of these nodes further branches into specific ingredients like "Wheat Grain", "Yeast", "Salt", and "Sugar". A "Scan Code" button is at the top right.
- Screenshot 3:** Shows a "Certificates" tab for the material. It lists two certificates: one from "Protect animals" and one from "Natural packaging", both issued by "Certificate Company" with a stake of 2 ETH and created "a day ago". A "Scan Code" button is at the top right.

Figura 33: Pagina de vizualizare a detaliilor unui material

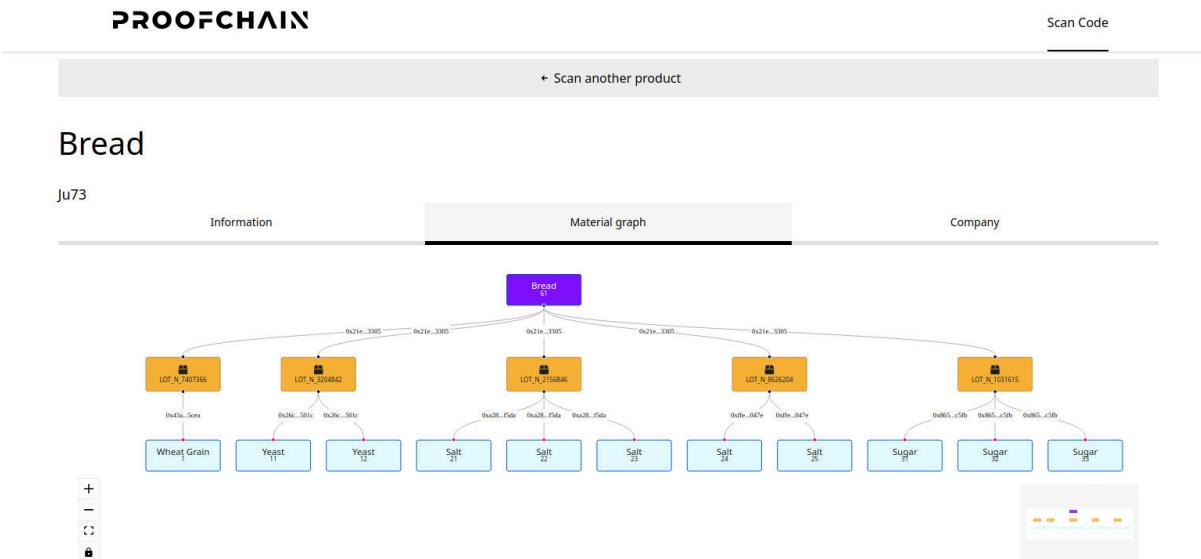


Figura 34: Structura arborescentă a unui material compus

Pentru a putea mai multe detalii, apasarea pe oricare element al graficului, deschide o fereastra cu informații suplimentare. În Figura 35 putem observa detaliiile după apăsarea pe nodul rădăcină, acestea fiind similare cu cele din Figura 34.

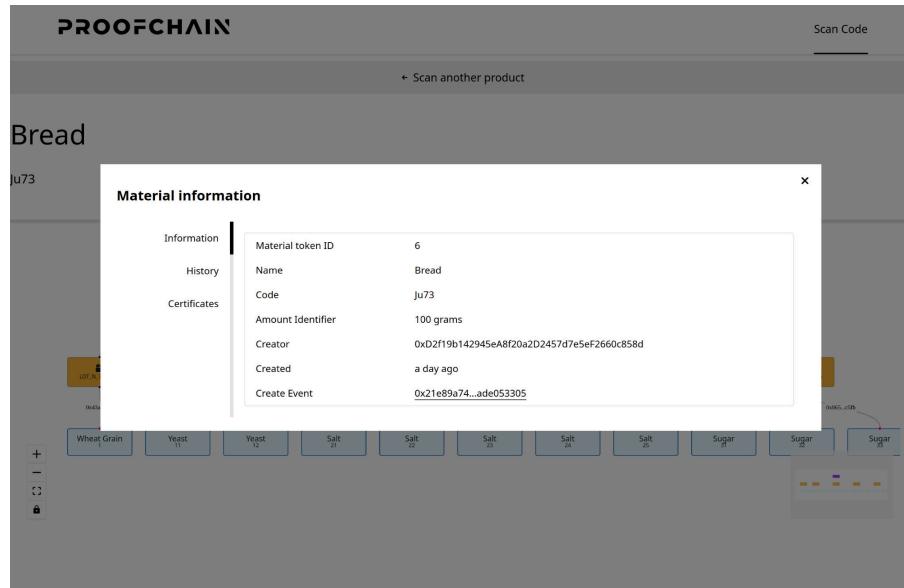


Figura 35: Fereastra deschisa după apăsarea pe nodul rădăcină

La apasarea pe un nod galben, echivalent al unui lot, o fereastră cu două secțiuni, precum în Figura 36. Prima conține detalii ale acestuia, iar a doua cu un istoric cu momentul când a fost creat, transporturile la care a avut parte sau momentul cand a fost șters.

Figura 36: Fereastra cu informațiile și istoricul unui lot

## 1.5 Integrarea cu soluțiile existente

Soluția pune la dispoziția utilizatorilor și o librărie de Javascript ce permite integrarea acestuia în diferite aplicații client. Aceasta oferă o funcționalitate completa a sistemului. De asemenea, acesta vine și cu o documentație cu privire la modul de folosire. Una dintre paginile documentației se poate vedea în Figura 37.

The screenshot shows a detailed API documentation page for the 'Company' module. At the top, there's a navigation bar with 'Proofchain library documentation' and search/filter options ('All', 'Inherited', 'Externals'). Below the header, the title 'Module Company' is displayed. The main content area is divided into several sections: 'Index', 'Enumerations' (listing 'CERTIFICATE\_ASSIGNMENT\_TYPE'), 'Classes' (listing 'Company'), 'Interfaces' (listing 'ICertificateAssignmentHistory', 'ICertificateInstance', 'ICompany'), 'Type aliases' (listing 'CompanyAssignedCertificateEvent', 'CompanyCanceledCertificateEvent', 'CompanyRevokedCertificateEvent', 'CompanyCreateEvent'), and a 'Exports' sidebar on the right listing various symbols like 'Base', 'Batch', 'CertificateAuthority', 'Company', etc. A code snippet for 'CompanyAssignedCertificateEvent' is shown at the bottom left, defining it as an object with properties: certificateAuthority (string), certificateCode (number), certificateInstanceId (number), companyAddress (string), and event (default).

Figura 37: Pagina de documentație a unei clase

## 1.6 Documentația utilizatorilor

Documentația utilizatorilor comunică și descrie modul în care soluția funcționează. Aceasta este împărțită în diferite secțiuni. Câteva dintre paginile acesteia sunt evidențiate în Figura 38 și Figura 39.

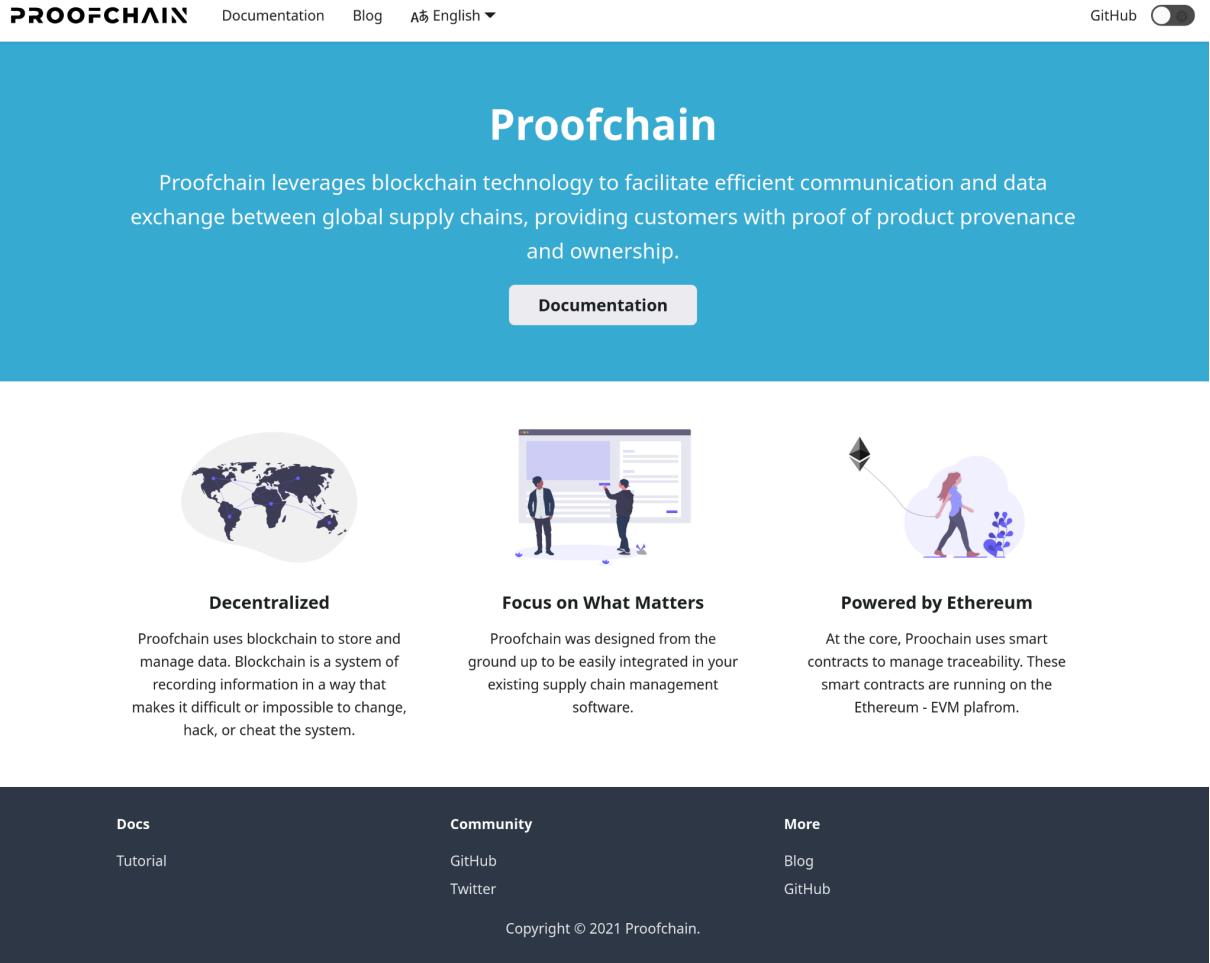


Figura 38: Pagina principala a documentației

**Intro**

Setting up  
Architecture  
Dashboard  
Library  
Client  
Smart Contracts

**Intro**

Proofchain leverages blockchain technology to facilitate efficient communication and data exchange between global supply chains, providing customers with proof of product provenance and ownership. It is a collection of smart contracts and Web3 based tools which are aiming to provide companies and customers with the required infrastructure to facilitate transparency across supply chains.

The following documentation is intended for getting a deeper understanding on how Proofchain works, both in terms of a technical point of view as well as from the point of view of its use in practice.

**At a glance**

Proofchain is a collection of blockchain based tools which make up an open platform that anyone can interact with. These include:

**Smart Contracts:** At the core of Proofchain are the smart contracts, written in Solidity, which live on the Ethereum blockchain. These make up an open platform that anyone can interact with.

**Proofchain Dashboard:** It offers a snapshot of the core features of the smart contracts based platform.

Figura 39: Pagina de introducere a documentației

## 1.7 Aplicația prezentatională

Aplicația prezentatională are rolul de a promova sistemul de trasabilitate atât pentru consumatori finali, cât și pentru companiile ce ar fi interesate de un produs care să rezolve problema transparentei în lantul de aprovisionare. Deoarece aplicația ar putea fi vizualizată de

un număr mare de utilizatori, inclusiv de persoane ce nu dețin cunoștințe în domeniul informaticii, aceasta adoptă un design modern pentru a transmite cu usurință mesajul și scopul sistemului. În Figura 40 se poate observa partea de început a acesteia.

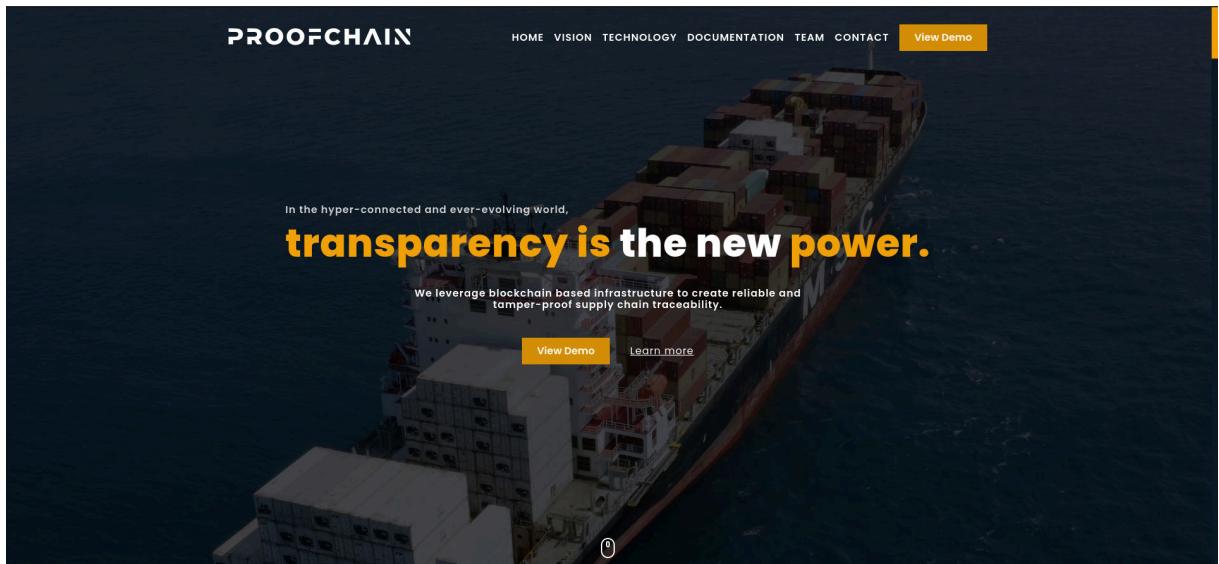


Figura 40: Pagina de start

## 1.8 Soluții similare

### 1.8.1 Provenance

Provenance [25], fondat în 2014 la Londra, este o companie software ce își dorește să ajute organizațiile să transparentizeze practicile de producție și impactul asupra mediului astfel încât să măreasca încrederea consumatorilor. Ideea principală este de a înregistra pe blockchain fiecare etapă din fabricarea unui produs și punerea acestor informații la dispoziția clienților.

### 1.8.2 Everledger

Everledger [46] este o companie ce utilizează blockchain pentru a înregistra tranzacțiile și certificatele obiectelor de lux precum diamantele sau vinurile. Caracteristicile pe care le identifică în mod unic precum înălțimea, lățimea, greutatea, culoarea etc sunt înregistrate. De asemenea aceștia folosesc și tehnologii din inteligența artificială și IoT pentru a îmbunătăți acuratetea informațiilor.

### 1.8.3 OriginTrail

OriginTrail [34] este un blockchain și un protocol open-source ce are ca scop rezolvarea problemelor de comunicare între organizațiile de tip logistic, precum informațiile fragmentate, interoperabilitate mica, vendor lock-in și asigurarea integrității datelor.

### 1.8.4 ShipChain

ShipChain [35] este un startup fondat în Los Angeles în 2017 ce se ocupă cu imbunatatirea trasabilității produselor, de la producție până la livrarea către client. Aceștia folosesc blockchain împreună cu senzori IoT pentru a monitoriza în timp real traseul și condițiile în care este transportată marfa.

### 1.8.5 Hyperledger Fabric

Hyperledger Fabric [36] este un produs al fundației Hyperledger și, deși acesta nu rezolva în mod direct o problema, este un blockchain modular și ușor configurabil, folosit de multe entități în optimizarea lanțului de aprovizionare. Printre capabilitățile sale se numără suport pentru EVM, izolarea datelor în canale securizate între entități, permisiuni de acces configurabile, diferiți algoritmi de consens. Deoarece Hyperledger are o vastă aplicabilitate în multe domenii, inclusiv trasabilitatea în lanțul de aprovizionare, companii precum AWS, Azure, IBM, Google, și Oracle pun la dispoziție instanțe configurabile ale acestei platforme.

# Capitolul 2 - Arhitectura aplicației

## 2.1 Contracte inteligente - Smart Contracts

La baza aplicației stau contractele inteligente, scrise în Solidity, ce se află pe platforma Ethereum. Acestea alcătuiesc un sistem cu care oricine poate interacționa. Figura 41 prezintă un sumar al celor mai importante contracte și al modului în care acestea comunică.

### 2.1.1 Arhitectura generală

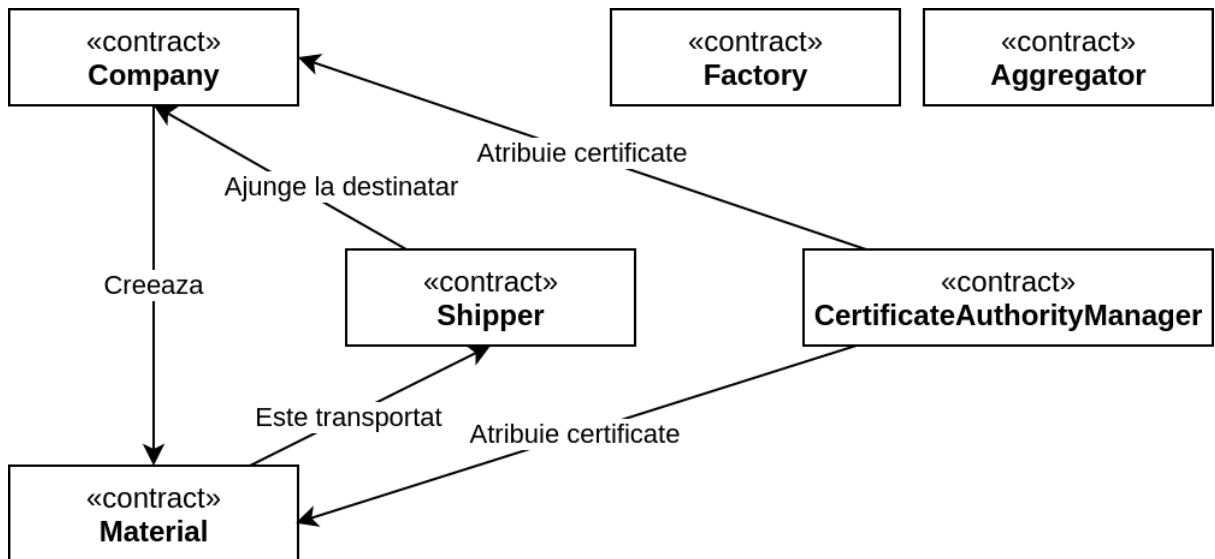


Figura 41: Arhitectura generală ale contractelor inteligente

La fiecare deployment al contractelor, contractul Factory este cel căruia îi va fi atribuită o adresa de tipul CA (Contract Account). În momentul încărcării pe blockchain, constructorul acestuia se va apela automat, iar contractele Aggregator, Company, Material și CertificateAuthorityManager se vor instanția. După instanțierea acestora, Factory va seta adresele lor în contractul Aggregator, contract ce va fi utilizat în alte contracte și va juca rolul unui proxy cu adresele tuturor instanțelor contractelor principale. Acest mod de organizare al codului este cunoscut în dezvoltarea contractelor inteligente sub numele de Factory Pattern, un contract ce instantiază alte contracte, inspirat din unul din principiile SOLID și anume Single Responsibility Principle. Principalul avantaj al acestei abordări este ca, pentru a accesa unul din contractele instantiatе, avem nevoie doar de adresa contractului Factory.

În Figura 1 din Anexa 3, este prezentată diagrama UML a întregii arhitecturi ale contractelor.

## 2.1.2 Autentificare

Autentificarea unei entități se realizează prin stocarea adresei publice în contractele inteligente. Pentru a crea o tranzacție, aceasta trebuie semnată cu ajutorul cheii private, ce nu ar trebui cunoscută decât de entitatea respectivă, asociate cheii publice. Astfel putem garanta faptul ca doar entitatea corecta poate efectua o tranzacție asociată cu o cheie publică din contract.

De asemenea, la anumite functii ce implica o preconditie pentru a putea fi executată, s-au folosit modificatori. Modificatorii (modifiers) sunt o metodă de a executa bucăți de cod înaintea executării unei funcții cu scopul de a efectua o verificare. Aceștia sunt inclusi în definiția funcției, iar dacă verificarea este cu success, se continua execuția. În Figura 42 este un exemplu al funcției de creare al unei instanțe de material, unde modificadorul “`senderIsTokenCreator`” se asigura ca doar cel ce a definit materialul poate crea o instanță a acestuia. De asemenea, deși nu este exemplificată în figura, pentru a crea o definiție al unui material, emițătorul tranzacției trebuie să dețină o companie asociată cheii publice al acestuia. Prin acest mod de inferență al verificărilor, ne putem asigura că sistemul este folosit corect din punct de vedere logic.

```
modifier senderIsTokenCreator(uint256 _materialTokenId) {
    require(
        msg.sender == materialToken[_materialTokenId].creator,
        "You are not the creator of this token"
    );
}
/**
 * Mint a new raw material
 *
 * @param _tokenID The material token id to be minted
 * @param _amount The amount of instances to be minted
 */
function mint(uint256 _tokenID, uint256 _amount) public senderIsTokenCreator(_tokenID) {
    ...
}
```

Figura 42: Un exemplu de modificador și de utilizare al acestuia

## 2.1.3 Contracte

### 2.1.3.1 Companii

Fiecare companie este identificată în mod unic prin adresa Ethereum al creatorului. Contractul principal care expune functionalitatile de administrare al companiei, este Company, extinzand două contracte abstracte utilitare, Shipper și Certifiable, precum în Figura 2 din Anexa 3.

Pentru fiecare operație ce este relevantă în contextul trasabilității lanțului de aprovisionare, se emite un eveniment. Emeterea de evenimentele poate fi asociată cu înregistrarea și salvarea unor loguri într-o bază de date. Principalul avantaj al evenimentelor este costul mai mic de stocare al informațiilor și ușurința cu care acestea pot fi filtrate.

Contractul Company oferă următoarele funcționalități:

- Crearea unei companii - funcția *create*
- Asignarea unui certificat - funcția *assignCertificate*
- Revocarea unui certificat
  - De către autoritatea de certificare - funcția *cancelCertificate*
  - De către administratorul de sistem - funcția *revokeCertificate*

De asemenea, acesta mai prezintă și anumite metode utilizare, folosite pentru a putea lua informații cu privire la certificate precum *getCompanyCertificateInstance* sau *getCompanyCertificatesInstanceIds*. Contractul mai mosteneste și alte două contracte abstracte, Certifiable și Shipper, ce conțin diferite metode pentru administrarea certificatelor și a transporturilor.

### Contractul Certifiable

Acest contract conține metode specifice asignării și revocării certificatelor, metode ce vor fi extinse în contractele ce mostenesc Certifiable, adică Company și Material. De asemenea, la fiecare asignare al unui certificat se creează o instanță nouă a unor structuri(struct) de date ce rețin informații cu privire la codul certificatului asignat și suma plătită ca și garanție. Modificatorul *payable* din funcția *assignCertificate*, precum în Figura 43, indică platformei Ethereum ca respectiva funcție poate primi o cantitate de ether, cantitate ce va fi stocată la nivelul adresei contractului.

```
abstract contract Certifiable is CertificateAuthorityManagerReferencer {  
    ...  
    function assignCertificate(uint256 _certificateCode) public payable {  
        ...  
    }  
    ...  
}
```

Figura 43: Contractul Certifiable - assignCertificate

### Contractul Shipper

Contractul Shipper conține toată logica din spatele administrației transporturilor. Deoarece o companie poate fi și de tip logistic, Company extinde Shipper. Mai multe detalii se regăsesc mai jos, în secțiunea Transport.

### 2.1.3.2 Materiale

Contractele principale ce sunt responsabile de administrarea materialelor sunt Material și MaterialBase. MaterialBase conține preponderent definiții de structuri, variabile globale și funcții utilitare, iar Material conține logica de creare, instanțiere, certificare și adaugare în loturi a materialelor. Deși acestea puteau fi combinate într-un singur contract, pentru o mai bună separare a codului, Material extinde MaterialBase. Figura 3 din Anexa 3 evidențiază structura acestora printr-o diagramă UML.

#### Crearea unui material

Crearea unui material este realizată prin funcția *create*, specifică contractului *Material*. O definiție al unui material conține urmatoarele campuri:

- *materialTokenId* - Un id unic fiecărui material;
- *name* - Numele materialului;
- *code* - Un cod arbitrar, ce ar putea avea ca scop asocierea identității digitale cu cea fizică;
- *creator* - Adresa ethereum a entității ce a creat definiția
- *certificateInstanceIds* - Un vector ce conține id-urile specifice asignării contractelor către material;
- *amountIdentifier* - Un identificator care să specifică unitatea de masură pentru o instanță a materialului;
- *recipeMaterialTokenId (optional)* - Un vector de id-uri ale materialelor din care este compus;
- *recipeMaterialAmount (optional)* - Un vector de cantități ale materialelor din care acesta este compus. Ordinea specificată trebuie să corespundă cu cea din parametrul *recipeMaterialTokenId*.

#### Crearea unei instanțe

Pentru a genera o instanță a acestuia funcția *mint* este folosită, cu două tipuri de definiții. În cazul în care materialul este în Figura 44, este apelul funcției ce creează o instanță a unui material compus. Pe lângă id-ul definiției materialului, aceasta primește doi vectori. Primul vector reprezintă id-ul loturilor folosite, al doilea vector, multidimensional, specifică id-urile instanțelor materialelor ce vor fi folosite în producția noii instanțe.

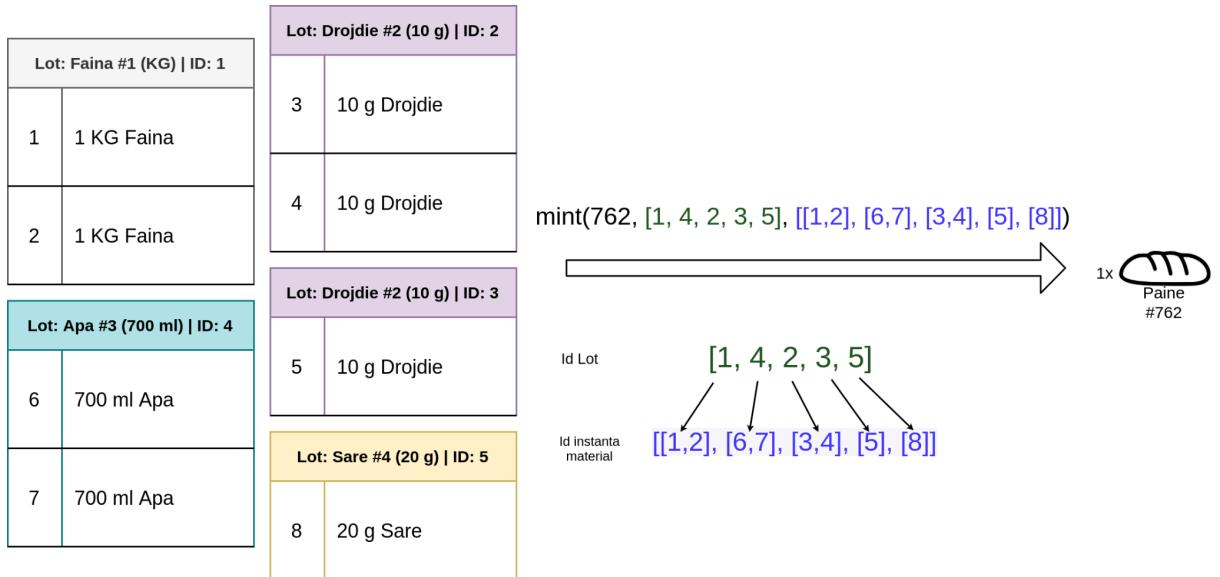


Figura 44: Asocierea materialelor și loturilor

După executarea cu success al unei funcții *mint*, materialele de intrare specificate sunt eliminate din posesia companiei, iar apoi se generează o instanță a noului material. Acesta este identificată cu ajutorul unui număr unic (uuid) pentru referirea ulterioară.

#### Adaugarea intr-un lot

Adaugarea instanțelor de materiale într-un lot se realizează apelând funcția *createBatch*, ce primește ca parametri de intrare următoarele date:

- (optional) Un cod arbitrar - reprezinta un identificator pentru lotul fizic
- Vector de uuid-uri ale instanțelor de materiale ce vor fi adăugate în lot

În momentul creării unui lot, acesta primește un id unic, diferit de codul arbitrar specificat în parametrul funcției. Lotul intră în posesia companiei creator, iar instanțele de materiale folosite sunt eliminate din posesia acesteia.

#### Certificarea

Certificarea materialelor este similară cu cea a companiilor. Contractul *MaterialBase* extinde *Certifiable*, iar *MaterialBase* este extins de *Material* ce adaugă funcționalitatile necesare certificării materialelor peste funcțiile existente din contractul *Certifiable*, *assignCertificate*, *cancelCertificate* și *revokeCertificate*.

#### 2.1.3.3 Loturi

Adaugarea instanțelor materialelor în loturi se realizează apelând funcția *createBatch* din contractul *Materials*. Aceasta conține informații cu privire la unicitatea lotului (*batchId*), proprietarul lotului (*owner*), un cod arbitrar (*code*), id-ul tipului de materiale ce sunt stocate

(materialTokenId) și uuid-urile fiecărei instanțe ale materialelor(materialsUuid). Funcția creeaza un lot, *createBatch*, primește doi parametri:

- Un cod arbitrar, cu scopul de a identifica lotul digital cu cel fizic;
- Un vector de numere, reprezentand uuid-urile instantelor materialelor ce vor fi adăugate în lot.

#### 2.1.3.4 Transport

La nivelul contractelor inteligente transportul reprezintă schimbul de proprietate între două entități, intermediar fiind compania de transport. Contractul Shipper extins de Company conține codul ce administrează transporturile, precum este evidențiat și în Figura 4 din Anexa 3. În Figura 45 este vizualizat modul în care este desfășurat un transport fictiv și în care sunt apelate diferite funcții din contracte.

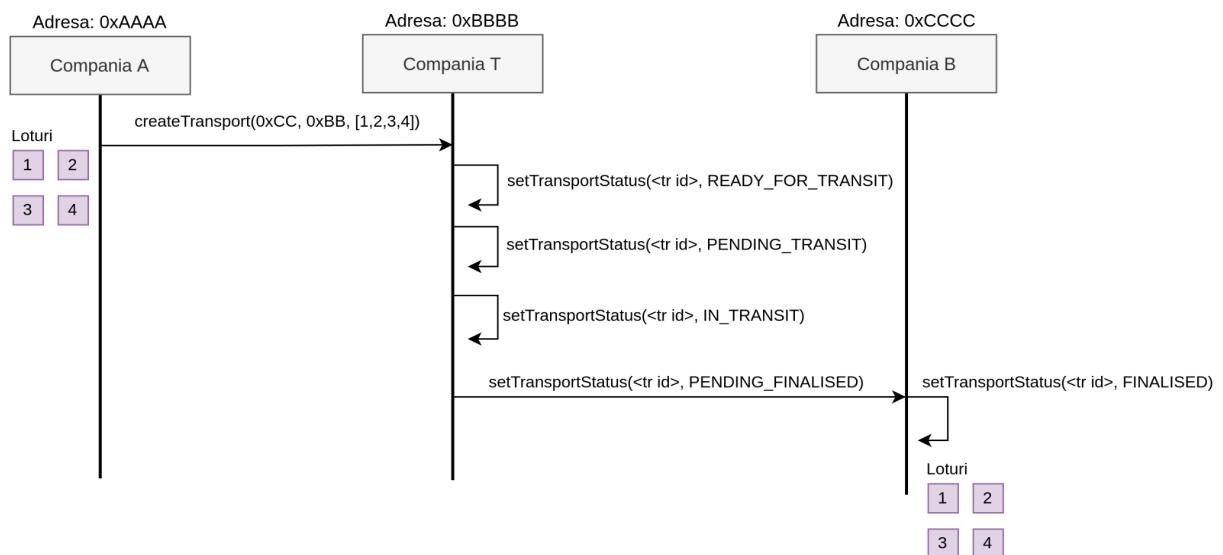


Figura 45: Fluxul unui transport fictiv

Pașii în realizarea unui transport sunt următorii:

1. Adresa specifică companiei ce deține loturile, va crea o tranzacție pentru a apela funcția *createTransport*, ce inițiază un transport, specificand adresa Ethereum a destinatarului, adresa Ethereum a companiei ce va intermedia transportul și un vector cu id-urile loturilor transportate. De asemenea aceasta funcție poate primi și un hash keccak256 a parolei.
2. Adresa specifică companiei de transport inițiază o serie de tranzacții pentru a apela funcția *setTransportStatus*, acestea setând anumite stări precum READY\_FOR\_TRANSIT, PENDING\_TRANSIT, IN\_TRANSIT, PENDING\_FINALISED, stări care semnifică evenimente a transportului fizic.
3. După ce compania de transport setează starea PENDING\_FINALISED, pentru a face schimbul de proprietate a loturilor către destinatar, adresa companiei destinatar trebuie

sa creeze o nouă tranzacție cu apelul funcției *finaliseTransport*, setand astfel transportul ca FINALISED. În cazul în care transportul a fost inițiat cu o parola, aceasta trebuie specificată în parametrii funcției.

### Schimbul de proprietate

Finalizarea unui transport implica un schimb de proprietate asupra loturilor. Stocarea deținătorilor loturilor se realizează în două moduri, primul este prin specificarea campului *owner* în structura lotului, iar al doilea este prin folosirea unui mapping precum în Figura 46. Acest lucru este motivat de faptul că fiecare operație realizată pe platforma Ethereum există un cost. În cazul în care am fi dorit să verificam dacă un lot aparține unei companii, iterarea fiecărei structuri nu ar fi fost cea mai eficientă variantă, complexitatea acestei operațiuni fiind O(n). Folosind un mapping, putem afla în O(1) dacă o adresa deține un anumit lot.

```
// true if an address has a specific batch id (address => (batchId => [true/false]))  
mapping(address => mapping(uint256 => bool)) public addressBatches;
```

Figura 46: Un mapping ce specifică dacă o adresa deține un lot

#### 2.1.3.5 Autoritatea de certificare

Autoritatea de certificare poate crea certificate și a le administra anumitor companii ce îndeplinesc specificațiile din acestea. Contractul *CertificateAuthorityManager* se ocupă cu crearea certificatelor și a entităților ce le pot asigna. Structura autoritatii de certificare este compusă dintr-un nume și adresa creatorului.

#### 2.1.3.6 Certificate

Certificatele reprezintă modalitatea prin care o companie sau un produs își transmite calitățile sau practicile de business.

Un certificat conține un cod unic, generat la creare, un nume și o descriere ce ar trebui să sumarizeze scopul aceluia certificat, tipul de certificat și creatorul acestuia. Tipurile de certificate se împart pe mai multe categorii precum:

- ENVIRONMENTAL\_IMPACT - categorie ce descrie impactul asupra mediului pe care îl are un material sau o companie;
- SAFETY\_AND\_QUALITY - categorie ce descrie siguranța utilizării unui material sau al procesului de producție al acestuia;
- HEALTH\_AND\_NUTRITION - categorie ce descrie beneficiile ingredientelor produselor sau modul în care acestea au fost fabricate;
- SOCIAL\_IMPACT - categorie descrie impactul social pe care îl are cumpărarea de produse sau practicile companiei;

- ANIMAL\_WELFARE - categorie ce descrie impactul pe care îl are produsul asupra animalelor sau practicile de creștere ale acestora;
- OTHER - alte categorii, la alegerea utilizatorului.

#### 2.1.4 Testarea contractelor

Testarea contractelor se realizeaza pe un blockchain local, acesta fiind pornit prin comanda *npm run ganache* din folderul rădăcină al proiectului. Sursa acestei comenzi se află în fișierul package.json, și porneste un blockchain Ethereum local la adresa 127.0.0.1:8545.

Testele unitare sunt împărțite pe categorii precum testarea functionalitatii codului specific companiilor, materialelor, loturilor, certificatelor, etc... SDK-ul Truffle pune la dispozitie un mediu preconfigurat, ce are la bază libraria Web3.js pentru interactionarea cu contractele și Mocha, un framework Javascript de testare.

#### 2.1.5 Încărcarea contractelor

Pentru ca aplicațiile client sa poată interacționa cu contractele inteligente, acestea trebuie încărcate într-un blockchain Ethereum public. Denumirea blockchain-ului folosit este Rinkeby. Conectarea cu acesta se realizeaza prin intermediul unui serviciu ce pune la dispozitie noduri conectate și pre-configurate, Infura [40]. De asemenea, încărcarea contractelor presupune o tranzacție și implicit un cost din partea unei adrese. Aceasta adresa este specificată în configurația truffle în fișierul truffle/truffle-config.js sub forma instanțierii unui obiect de tipul Web3.js Provider folosind adresa privată și adresa nodului Ethereum unde se va trimite tranzacția

## 2.2 Proofchain.js Library - Librăria pentru integrare

Libraria de Javascript permite integrarea sistemului în diferite aplicații client. Aceasta este scrisă în Typescript, compilată mai apoi în Javascript, lucru ce aduce multe beneficii atât în procesul de dezvoltare cât și de integrare. Cateva dintre beneficiile utilizatii Typescript sunt:

- Tipurile de date statice (static typing)
- Integrare cu diferite IDE-uri și autocompletare
- Eleganta codului

Deoarece scopul sistemului de trasabilitate este acela de a fi integrat ca o extensie în sistemele de management al lanțului de aprovizionare, libraria vine în ajutorul viitorilor dezvoltatori pentru a ușura integrarea. Lucrul direct cu contractele inteligente presupune, pe

länga cunoștințele limbajului de programare, și cunoștințe adiționale specifice platformei Ethereum și a librăriei Web3.js, iar libraria își propune să rezolve aceasta problema abstractizând modul de interactionare cu contractele.

### 2.2.1 Instalare

Libraria poate instala cu folosind managerul de module NPM, folosind comanda `npm install proofchain-library`. Aceasta va crea un folder `node_modules` unde, printre alte dependințe, va exista și un director cu codul sursa al librăriei.

### 2.2.2 Inițializare

Înainte de a putea fi utilizată, libraria trebuie inițializată cu o anumită configurație, pentru a putea interacționa cu contractele inteligente.

Parametrii de configurare ai librăriei sunt specificați la crearea unei noi instanțe. Aceștia sunt:

- *Web3* - o instanță a `web3.js`, instanță ce trebuie să contină setata conexiunea cu un nod Ethereum și un portofel;
- *fromAddress* - adresa de pe care vor fi trimise tranzacțiile;
- *factoryContractAddress* - adresa contractului Factory încărcata pe blockchain.

În momentul instantierii librăriei, aceasta apelează contractul Factory pentru a lua adresa contractului Aggregator, contract ce conține adresele pentru Company, Material și CertificateAuthority. Acest lucru reprezinta un avantaj pentru utilizatorul librăriei, deoarece prin specificarea unei singure adrese la instanțierea librăriei, el are acces la toate functionalitatile sistemului de trasabilitate chiar dacă acestea sunt împărtășite în mai multe contracte.

De asemenea, libraria poate fi inițializată și fără parametrul *fromAddress* sau portofelul configurat cu `Web3.js`, fapt ce asigură lucrul doar cu metodele care nu modifică starea blockchain-ului, în cazul în care avem nevoie doar de luat date din contracte.

### 2.2.3 Integrarea cu Web3.js

Pentru comunicarea cu nodul Ethereum și implicit cu contractele inteligente, libraria folosește `Web3.js`. `Web3.js` este o librărie, sau mai bine zis o colecție de librării mai mici, ce facilitează crearea tranzacțiilor și semnarea acestora. De asemenea `Web3.js` este utilă și pentru utilizarea portofelelor virtuale (transferare ether, interogare balanță).

## 2.2.4 Integrarea cu contractele inteligente

Libraria pune la dispoziție funcționalitatea completă a contractelor inteligente ale sistemului. Fiecare funcționalitate din contracte se regăsește ca o metodă în librărie, astfel încât utilizatorii ce ar dori o integrare cu sistemul de trasabilitate nu necesită și alte dependințe tehnice.

## 2.2.5 Funcționalități specifice contractelor

În structura librăriei, fiecare entitate al sistemului este reprezentată printr-o clasa. Cateva dintre clasele principale sunt:

- Company (Company.ts) - Funcții specifice administrării companiilor
- Material (Material.ts) - Funcții specifice administrării materialelor
- Batch (Batch.ts) - Funcții specifice administrării loturilor
- Transport (Transport.ts) - Funcții specifice administrării transporturilor
- CertificateAuthority (CertificateAuthority.ts) - Funcții specifice administrării autorităților de certificare

Fiecare dintre aceste clase sunt instantiate la momentul instantierii librăriei. Un element esențial pe care aceste clase îl folosesc este obiectul ABI (Application Binary Interface) al contractelor asociate cu care comunica.

Pentru stabilirea numărului de unități de gas al fiecărei tranzacții, libraria folosește funcționalitatea de estimare al platformei Ethereum, ce, prin metoda RPC “eth\_estimateGas” executa tranzacția în nodul conectat dar nu o include în blockchain, și returnează numărul de unități de gas folosite. Astfel se poate stabili cu precizie cantitatea exactă de gas utilizat.

## 2.2.6 ABI (Application Binary Interface)

Obiectul de tip ABI este generat, împreună cu bytecode-ul, la momentul compilării fiecărui contract. Aceste descrie antetul fiecărei funcții din contract (tipul de funcție, modificatori, numele fiecărui parametru, tipul de date, ce returnează funcția). Pentru apelarea metodelor din contractele inteligente, Web3.js are nevoie de adresa respectivului contract și de ABI-ul acestuia, un obiect JSON. Figura 47 prezintă un exemplu al modul de interacțiune dintre libraria sistemului, Web3.js și ABI-ul contractelor, exemplul fiind pentru Company.

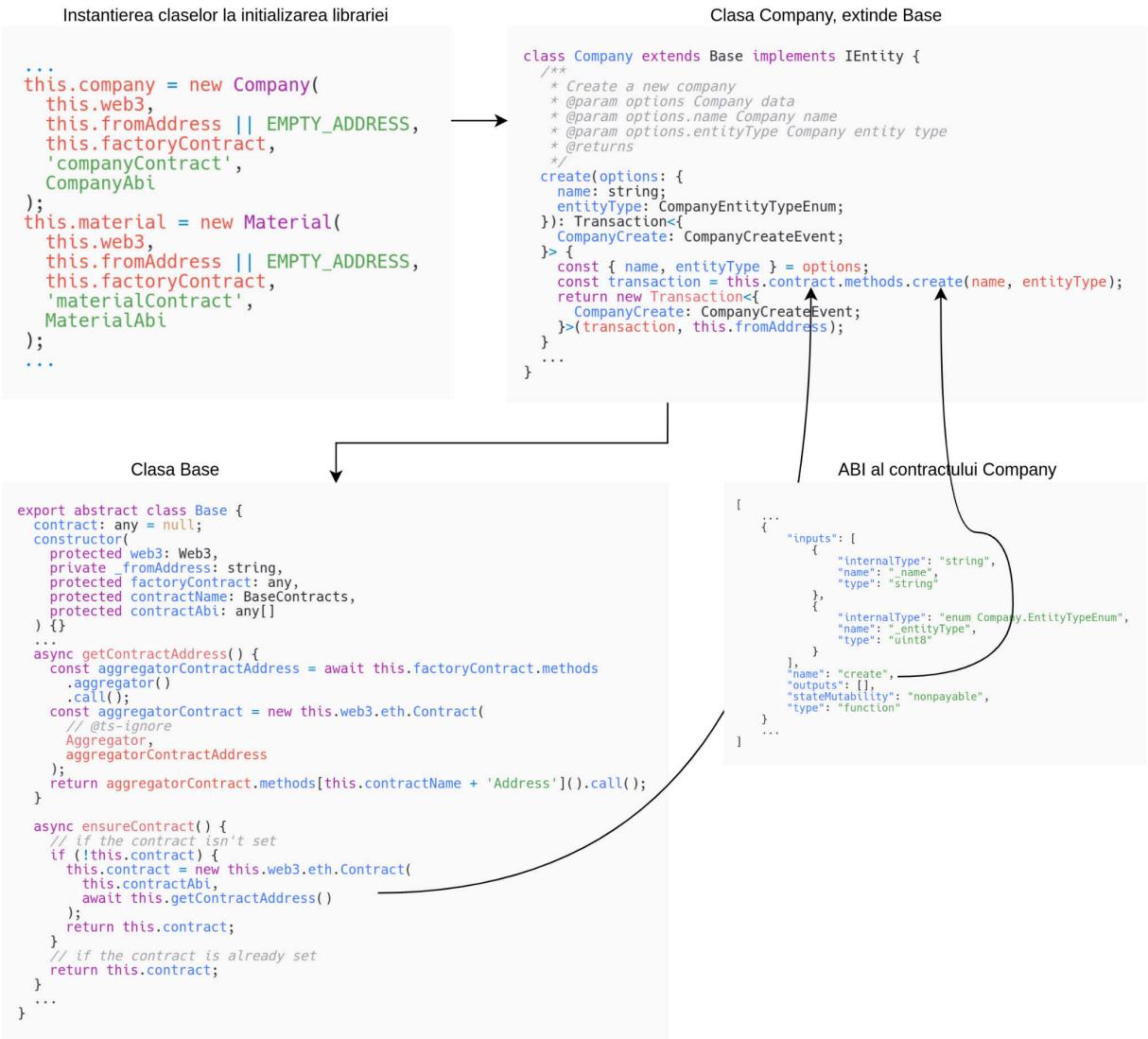


Figura 47: Un mod scalabil al integrării cu Web3 și cu contractele inteligente

Clasa Base este folosită de toate celelalte clase ce au asociate contracte inteligente.

## 2.2.7 Modificarea costului tranzacției

În cazul unei tranzacții, libraria pune la dispoziție un parametru esențial din componenta unei tranzacții, costul per unitate de gas (gas cost). La emiterea unei tranzacții, nodul Ethereum conține o coadă de alte tranzacții ce urmează a fi procesate și adăugate într-un bloc. Ordinea acestor tranzacții este data de prețul pe care emițătorul este dispus să îl plătească, iar cu cat prețul, cantitatea de ether per unitate de gas este mai mare, cu atât tranzacția va fi procesată mai repede.

Libraria permite modificarea prețului per unitate de gas al unei tranzacții precum este exemplificat în Figura 48, prin specificarea acestuia în Wei, ceea mai mică unitate de ether.

```

await proofchainInstance.company
.create({
  name: "Company name",
  entityType: CompanyEntityTypeEnum.MANUFACTURER
})
.setGasPrice(200000000000)
.send();

```

Figura 48: Crearea unei companii cu un cost modificat

În cazul în care nu este setat acest preț, se va folosi comanda RPC “eth\_gasprice” ce returnează prețul mediul al tranzacțiilor din ultimele blocuri pentru a seta aceasta valoare.

## 2.2.8 Testarea librăriei

Toate metodele librariei sunt testate cu ajutorul mediului de testare Jest și al blockchain-ului local Ganache. Înainte de fiecare rulare a testelor, un blockchain local este instantiat iar contractele sunt încărcate pe acesta, astfel asigurându-se un mediu de testare deterministic de fiecare data.

## 2.2.9 Documentarea librăriei

Datorită modului riguros de scriere al codului impus de Typescript și comentarea acestuia cu adnotatii specifice, documentarea este ușor de realizat. Pentru aceasta s-a folosit TypeDoc, un generator de documentatie API al codului Typescript. Acesta genereaza o aplicatie web interactiva ce conține informații cu privire la diferitele clase, metode, interfețe, enumeratii etc.. Figura 49 prezinta o pagina generata pe baza comentariilor din cod,

The screenshot displays the 'Proofchain library documentation / Company / Module Company' page. The left sidebar contains an 'Index' with sections for 'Enumerations', 'Classes', 'Interfaces', and 'Type aliases'. The 'Exports' sidebar on the right lists various TypeScript symbols and events. The 'Type aliases' section details three specific event types: CompanyAssignedCertificateEvent, CompanyCanceledCertificateEvent, and CompanyRevokeCertificateEvent.

Figura 49: Exemplu de documentație

## 2.3 Proofchain Web Dashboard

Panoul de administrare, Proofchain Web Dashboard, a fost dezvoltat pentru a ilustra atât funcționalitățile sistemului propus, cat și functionalitatile libreriei, acesta utilizând în totalitate libraria Proofchain.js pentru a realiza tranzacțiile specifice trasabilității. În termenii din domeniul calculului distribuit, acesta este o aplicație descentralizată (DApp), deoarece partea de backend rulează într-o rețea descentralizată de noduri. Partea de frontend este dezvoltată folosind Next.js împreună cu React.js și este compusă dintr-o serie de componente specifice libreriei React.js.

### 2.3.1 Autentificarea

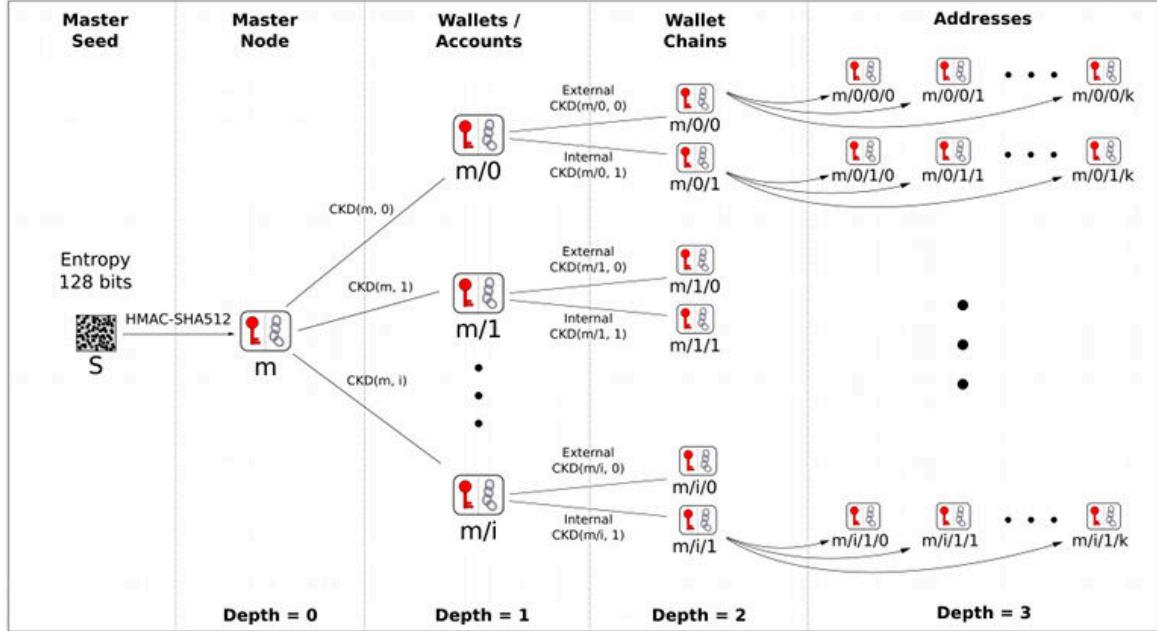
Autentificarea este realizată prin 3 moduri: extensia Metamask, Torus și fraza mnemonică.

Metamask injectează o instanță a libreriei Web3.js în lista variabilelor globale ale browser-ului, instanță ce este conectată cu un nod Ethereum și cu un wallet, iar pentru a trimite o tranzacție este necesară confirmarea acesteia în extensia browser-ului.

Torus este folosit pentru a facilita autentificarea cu contul Google. Principalul avantaj al serviciului Torus este acela că în momentul autentificării, fiecărui cont de utilizator îi este asociată o cheie privată unică cu care acesta poate semna tranzacțiile. Modul de stocare al cheilor private este non-custodial, ceea ce înseamnă că nimeni în afară de utilizator nu poate afla cheia privată asociată contului sau. Pe scurt, arhitectura acestui serviciu este bazată pe generarea de chei într-un mod distribuit (distributed key generation - DKG) și stocarea acestora folosind la baza Shamir's Secret Sharing [44], un mod de a stoca un mesaj secret în mai multe bucăți al acestuia.

Fraza mnemonică împreună cu un portofel deterministic ierarhic (HD Wallet) este o alternativă mai facilă pentru memorarea unor chei private. Fraza mnemonică este formată din 12 sau 24 de cuvinte ce generează un seed folosit de portofelul deterministic ierarhic pentru crearea unor chei private, bazată pe o cale de derivare (derivation path), precum este evidențiat și în Figura 50.

## BIP 32 - Hierarchical Deterministic Wallets



Child Key Derivation Function  $\sim CKD(x, n) = HMAC-SHA512(x_{Chain}, x_{PubKey} \parallel n)$

Figura 50: Exemplificare al arborelui de derivare, bazat pe diferite cai de derivare [45]

În contextul aplicației, lungimea frazei mnemonice este de 12 cuvinte, iar calea de derivare se poate schimba în cazul în care utilizatorul dorește acest lucru. Autentificarea se realizează prin afișarea unui formular, precum în Figura 51.

```

import * as bip39 from 'bip39';
import { hdkey } from 'ethereumjs-wallet';
import Web3 from 'web3';

export const getAccountFromMnemonic = async (
  mnemonic,
  derivationPath = "m/44'/60'/0'/0/0"
) => {
  // create seed from mnemonic phrase
  const seed = await bip39.mnemonicToSeed(mnemonic);
  // initialize HD Wallet
  const hdwallet = hdkey.fromMasterSeed(seed);
  // generate wallet from seed and derivation path
  const wallet = hdwallet.derivePath(derivationPath).getWallet();
  // get private key
  const privateKey = wallet.privKey.toString('hex');
  // generate web3js account object with private key
  return new Web3().eth.accounts.privateKeyToAccount(privateKey);
};

```

The screenshot shows the Proofchain application interface. On the left, there is a form for entering a mnemonic phrase (e.g., "stage analyst reform dune educate throw exile disagree pause search crouch finger") and a derivation path (e.g., "m/44'/60'/0'/0/0"). A checkbox "I've copied it somewhere safe." is present, and a "Import" button is at the bottom. An arrow labeled "Apelare" (Call) points from the "Import" button to the provided JavaScript code on the right, which demonstrates how to use the bip39 library to generate an account from a mnemonic phrase and a derivation path.

Figura 51: Autentificarea și generarea unui cont cu ajutorul frazei mnemonice și caii de derivare

### 2.3.2 Comunicarea cu blockchain

Panoul de administrare folosește libraria Proofchain pentru a comunica cu diferitele funcții ale contractelor inteligente. Aceasta este instanțiată pe baza unui fișier de configurare `.env` ce contine adresa contractului principal (Factory). De asemenea, în cazul în care utilizatorul este un client, acesta nu se va autentifica iar libraria va fi instantiată fără o cheie privată.

### 2.3.3 Administrarea stării

Pentru a stoca și modifica diferite date temporare, aplicația folosește React Redux, o librărie ce pune la dispoziția dezvoltatorilor un mod de administrare al datelor centralizat și scalabil.

## 2.4 Proofchain Web Client

Aplicația web pentru consumatori, Proofchain Web Client, a fost dezvoltată pentru a furniza o modalitate ușoară utilizatorilor de a vedea detalii specifice unui produs, precum materialele utilizate, certificatele materialelor utilizate, istoricul materialelor utilizate, loturile din care acestea au făcut parte, istoricul produsului și detalii despre compania ce a fabricat produsul.

Aceasta aplicatie este o extensie al panoului de administrare, Proofchain Web Dashboard. Din punct de vedere arhitectural, multe dintre componentele vizuale sunt refolosite, iar managementul stării este realizat în același context global al panoului de administrare. O diferență importantă este abilitatea consumatorilor de a interactiona cu aplicația fără nevoie de autentificare cu o pereche de chei criptografice, deoarece aceștia pot doar citi datele de pe blockchain, acțiune ce nu necesită semnarea unei tranzacții. În momentul introducerii unui uuid al unei instanțe de material, aplicația se folosește de log-urile generate de diferitele interacțiuni cu contractele inteligente pentru a căuta informații.

Pentru selectarea materialului dorit, utilizatorii pot scana un cod QR, reprezentand uuid-ul instanței materialului, sau îl pot introduce într-un formular. Libraria *qr-scanner* [47] facilitează lucrul cu camera dispozitivului și decodarea imaginii QR.

# Capitolul 3 - Scenarii de utilizare

Următoarele scenarii descriu modul în care un utilizator ar putea interacționa cu sistemul.

## 3.1 Crearea identității digitale ale unei companii

**Actor principal:** companie sau autoritate de certificare

**Preconditii:**

1. Utilizatorul este autentificat
2. Adresa folosită pentru autentificare conține suficient ether pentru efectuarea tranzacției

**Postcondiții:**

1. Identitatea virtuală a fost creată
2. Utilizatorul poate interacționa cu restul platformei

**Scenariul principal:**

1. Utilizatorul selectează tipul de identitate, companie sau autoritate de certificare
2. După selectare, acesta trebuie să introducă numele și tipul de companie într-un nou formular
3. Înainte de trimiterea tranzacției către un nod, aplicația notifică utilizatorul de costurile asociate acesteia
4. După trimiterea și confirmarea tranzacției, aplicația redirectionează utilizatorul către pagina principală

**Scenarii alternative:**

5. Datele introduse nu sunt valide
  - a. Aplicația va afișa un mesaj de eroare
6. Utilizatorul nu are o cantitate suficientă de ether
  - a. Aplicația va afișa un mesaj de eroare în momentul în care tranzacția este revocată de nod
7. Sistemul întâmpină o problemă
  - a. Aplicația va afișa un mesaj de eroare

În Figura 52 este evidențiat modul în care decurge crearea unei identități de tipul companie.

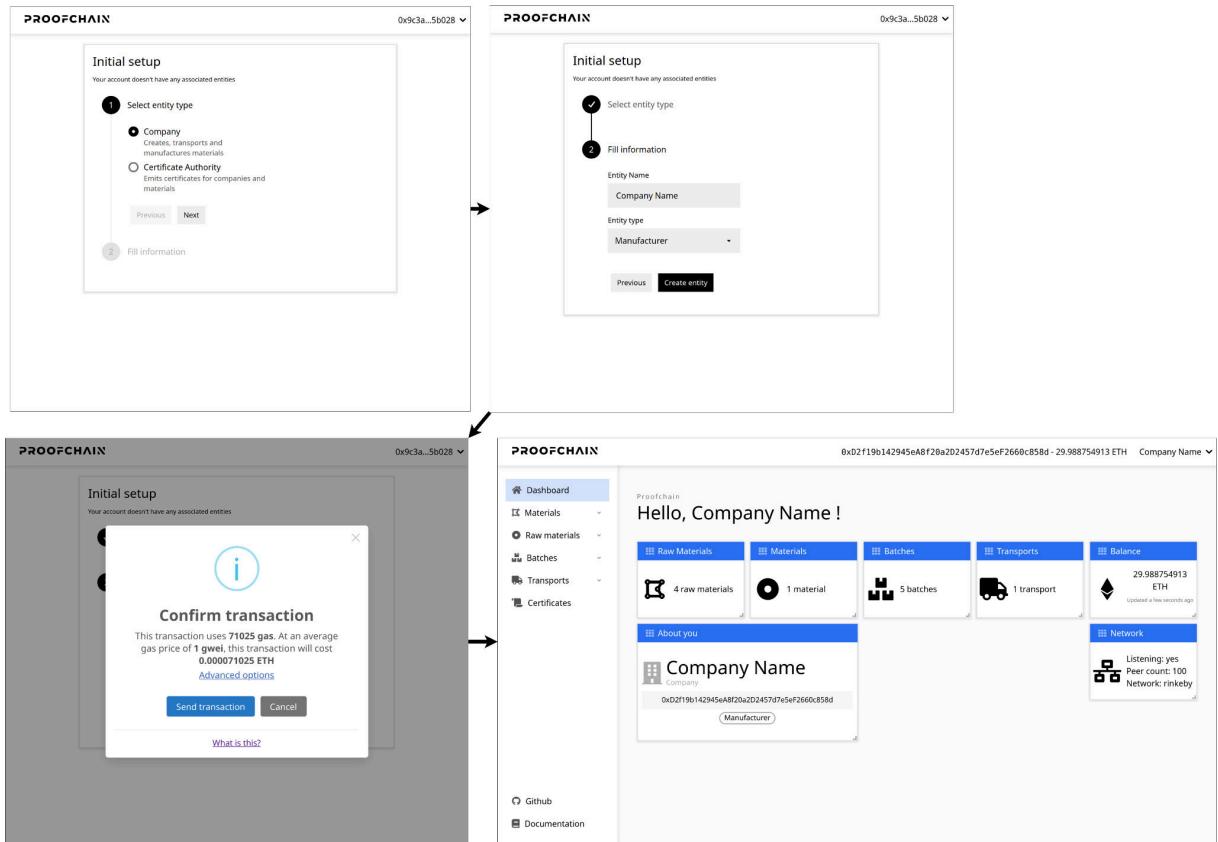


Figura 52: Interfața pentru crearea unei identități.

### 3.2 Crearea unui material compus

#### Preconditii:

1. Utilizatorul este autentificat
2. Adresa folosită pentru autentificare conține suficient ether pentru efectuarea tranzacției
3. Utilizatorul are la dispoziție id-urile materialelor simple din care va fi compus cel nou

#### Postconditii:

1. Materialul compus a fost creat după specificatii

#### Scenariul principal:

1. Utilizatorul introduce detaliile specifice materialului precum numele, codul, identificatorul de cantitate al unei instanțe și celealte materiale din care acesta este compus
2. Înainte de trimiterea tranzacției către un nod, aplicația notifică utilizatorul de costurile asociate acesteia
3. După trimiterea și confirmarea tranzacției, aplicația redirectionează utilizatorul către pagina principala

În Figura 53 este evidențiată pagina de creare al unui material.

The screenshot shows the PROOFCHAIN application's 'Create Material' page. The left sidebar has a 'Materials' section with 'All materials' and 'Create material' selected. The main form has fields for 'Material name' (Bread), 'Descriptive material name', 'Material code' (BHYA3), 'Optional material identification code', 'Amount identifier' (Kilogram - kg), and two rows for 'Material Token Id' (0 and 1) with their respective amounts (1). A 'Create material' button is at the bottom.

Figura 53: Pagina de creare al unui material

#### Scenarii alternative:

4. Datele introduse nu sunt valide
  - b. Aplicația va afișa un mesaj de eroare
5. Utilizatorul nu are o cantitate suficientă de ether
  - a. Aplicația va afișa un mesaj de eroare în momentul în care tranzacția este revocată de nod
6. Sistemul întâmpină o problemă
  - a. Aplicația va afișa un mesaj de eroare

### 3.3 Crearea unei instanțe a unui material compus

#### Preconditii:

1. Utilizatorul este autentificat
2. Adresa folosită pentru autentificare conține suficient ether pentru efectuarea tranzacției
3. Utilizatorul are la dispoziție id-ul materialului caruia ii va fi creata o instanta
4. Utilizatorul este creatorul definiției materialului caruia ii va fi creata o instanta

- Utilizatorul deține loturi ce conțin instanțe ale materialelor ce vor utiliza ca ingrediente

#### **Postconditii:**

- O instanță a materialului a fost generată
- Materialele folosite vor fi eliminate din loturi

#### **Scenariul principal:**

- Utilizatorul introduce detaliile în legătura cu ce materiale vor fi folosite. Aceste detalii conțin id-ul lotului și uuid-ul materialul din acel lot ce va fi folosit. Pentru fiecare ingredient trebuie specificat acest lucru.
- Înainte de trimiterea tranzacției către un nod, aplicația notifică utilizatorul de costurile asociate acesteia
- După trimiterea și confirmarea tranzacției, aplicația reîncarcă datele din pagina pentru a afișa instanța materialului nou creat

Figura 54 prezintă pagina de creare al unei instanțe, cu datele ingredientelor completate.

Material Token Id	Material name	Amount
0	Wheat	1 kg
1	Water	1 l

UUID	Material ID	Mint Transaction
7	2	0xadcb9597...65f60de6e

Figura 54: Crearea unei noi instanțe a unui material

#### **Scenarii alternative:**

- Loturile sau instanțele materialelor folosite nu există
  - Aplicația va afișa un mesaj de eroare
- Datele introduse nu sunt valide
  - Aplicația va afișa un mesaj de eroare
- Utilizatorul nu are o cantitate suficientă de ether
  - Aplicația va afișa un mesaj de eroare în momentul în care tranzacția este revocată de nod
- Sistemul întâmpină o problemă

- a. Aplicația va afișa un mesaj de eroare

### 3.4 Crearea unui transport către o altă entitate

#### **Preconditii:**

1. Utilizatorul este autentificat
2. Adresa folosită pentru autentificare conține suficient ether pentru efectuarea tranzacției
3. Utilizatorul are la dispoziție adresa ethereum a destinatarului
4. Utilizatorul are la dispoziție adresa ethereum a entitatii logistice ce va intermedia transportul
5. Utilizatorul deține loturile ce vor fi transportate

#### **Postconditii:**

1. Loturile vor fi eliminate din posesia expeditorului
2. Un nou transport va fi creat

#### **Scenariul principal:**

1. Utilizatorul introduce adresa ethereum al entitatii logistice și al destinatarului împreuna cu id-urile loturilor ce vor fi transportate
2. Înainte de trimitera tranzacției către un nod, aplicația notifica utilizatorul de costurile asociate acesteia
3. După trimitera și confirmarea tranzacției, aplicația reîncarcă datele din pagina pentru a afișa instanța materialului nou creat

#### **Scenarii alternative:**

1. Loturile sau adresele entităților nu există
  - a. Aplicația va afișa un mesaj de eroare
2. Datele introduse nu sunt valide
  - a. Aplicația va afișa un mesaj de eroare
3. Utilizatorul nu are o cantitate suficientă de ether
  - a. Aplicația va afișa un mesaj de eroare în momentul în care tranzacția este revocată de nod
4. Sistemul întâmpină o problemă
  - a. Aplicația va afișa un mesaj de eroare

### 3.5 Vizualizarea istoricului unui material de către consumator

#### **Preconditii:**

- Utilizatorul este pe pagina corespunzătoare în aplicația web

#### **Postconditii:**

- Utilizatorul poate interacționa cu informațiile

#### **Scenariul principal:**

- Utilizatorul introduce uuid-ul instanței materialului

În Figura 55 este evidențiată una din paginile de vizualizare al istoricului produsului.

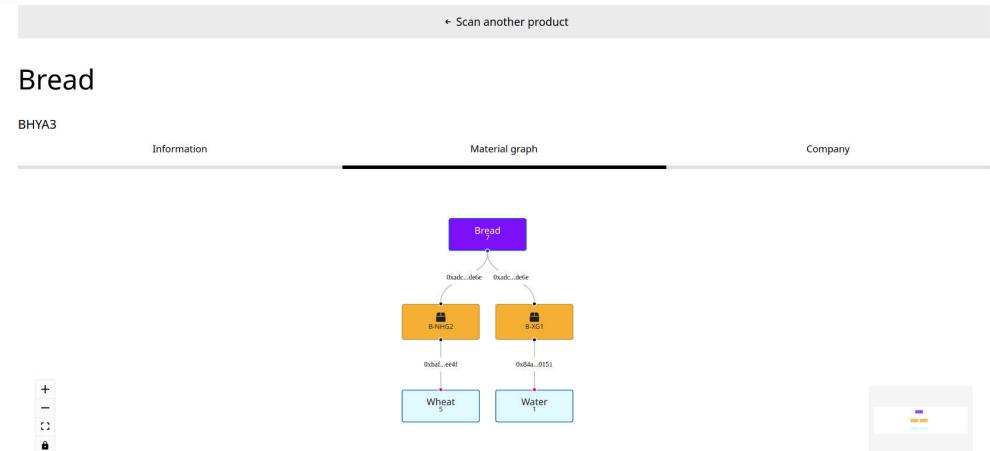


Figura 55: Graficul istoricului unui produs

#### **Scenarii alternative:**

- Uuid-ul instanței materialului nu există
  - Aplicația va afișa un mesaj de eroare
- Datele introduse nu sunt valide
  - Aplicația va afișa un mesaj de eroare
- Utilizatorul nu are o cantitate suficientă de ether
  - Aplicația va afișa un mesaj de eroare în momentul în care tranzacția este revocată de nod
- Sistemul întâmpină o problemă
  - Aplicația va afișa un mesaj de eroare

## Capitolul 4 - Concluzii și îmbunătățiri viitoare

Primul obiectiv a fost dezvoltarea unui sistem de trasabilitate care să ilustreze aplicabilitatea tehnologiei blockchain în domeniul lanțurilor de aprovizionare. Prin prezenta lucrare am ilustrat metodele prin care am implementat un sistem cu diverse capabilități în această direcție.

Un obiectiv secundar, a fost studiul tehnologiei blockchain. Cunoștințele despre acest subiect au fost dobandite atât din studiul resurselor bibliografice ale lucrării, din alte surse de pe Web cat și din experiența dezvoltării acestui proiect.

Potențialul de dezvoltare pentru aplicația prezentată este reprezentat de mai multe planuri.

În primul rand, folosind un blockchain public, fiecare interacțiune din interiorul aplicației este vizibilă de către oricine. Acest fapt ar putea pune anumite companii, ce ar folosi acest sistem de trasabilitate, într-un dezavantaj competitiv față de alte companii. O posibilă soluție ar putea fi utilizarea unei arhitecturi hibride, ce ar permite entităților lanțului de aprovizionare să comunice într-un blockchain privat, iar hash-ul acestei comunicatii sau ale anumitor tranzacții să fie stocat pe un blockchain public.

În al doilea rand, din punct de vedere funcțional, pentru a modela procesele de creare și transport ale materialelor, s-a utilizat o metodă generică ce este posibil să nu poată fi folosită pentru anumite procese complexe. Aceasta va trebui adaptată în funcție de nevoile fiecărei entități.

În al treilea rând, cu ajutorul contractelor inteligente și al unui proces de tranzacționare similar cu aranjamentul contractual de tip *escrow*, plata produselor transportate poate fi efectuată pe blockchain. Prin acest mod entitățile pot beneficia de un schimb monetar direct și sigur, fără un număr mare de intermediari.

În al patrulea rând, o posibilă îmbunătățirea substanțială ar putea fi și includerea domeniului IoT în procesul de transparentizare și trasabilitate. Introducerea unor senzori ce comunică cu diferite contracte inteligente ar putea aduce un avantaj al siguranței pentru consumatori, cat și unul competitiv pentru companii. De asemenea, în cazul în care un sistem de tip *escrow* ar fi existent, se pot implementa la nivelul contractelor anumite reguli cu privire la valori ale senzorilor, iar în cazul în care acestea nu sunt respectate, anumite penalizări pot fi suportate de entitățile logistice.

# Bibliografie

- [1] Liao, Y., Loures, E. R., Deschamps, F., Brezinski, G., & Venâncio, A. (2017). The impact of the fourth industrial revolution: a cross-country/region comparison. *Production*, 28, e20180061. DOI: 10.1590/0103-6513.20180061. URL:  
<https://www.scielo.br/pdf/prod/v28/0103-6513-prod-28-e20180061.pdf>
- [2] Klaus Schwab, “The Fourth Industrial Revolution: what it means, how to respond” (2016). URL:  
<https://www.weforum.org/agenda/2016/01/the-fourth-industrial-revolution-what-it-means-and-how-to-respond/>
- [3] James Manyika, Michael Chui, Jacques Bughin, Richard Dobbs, Peter Bisson, Alex Marrs, “Disruptive technologies: Advances that will transform life, business, and the global economy”, McKinsey Global Institute, May 2013. URL:  
[https://www.mckinsey.com/~media/McKinsey/Business%20Functions/McKinsey%20Digital/Our%20Insights/Disruptive%20technologies/MGI\\_Disruptive\\_technologies\\_Executive\\_summary\\_May2013.pdf](https://www.mckinsey.com/~media/McKinsey/Business%20Functions/McKinsey%20Digital/Our%20Insights/Disruptive%20technologies/MGI_Disruptive_technologies_Executive_summary_May2013.pdf)
- [4] Hussain A.H Awad, Mohammad Othman Nassar, A Broader view of the Supply Chain Integration Challenges, in International Journal of Innovation, Management and Technology, Vol. 1, No. 1, April 2010, ISSN: 2010-0248,  
URL:[https://www.kau.edu.sa/Files/0056839/Researches/57869\\_28319.pdf](https://www.kau.edu.sa/Files/0056839/Researches/57869_28319.pdf)
- [5] Comisia Comunităților Europene, REGULAMENTUL (CE) NR. 93/2005, 19 ianuarie 2005. URL:  
<https://eur-lex.europa.eu/legal-content/RO/TXT/PDF/?uri=CELEX:32005R0093&from=EN>
- [6] Futerra, The Consumer Goods Forum, The Chartered Institute of Marketing, “The Honest Product”, July 2018. URL:  
<https://www.theconsumergoodsforum.com/wp-content/uploads/2018/10/CGF-Futerra-Transparency-and-the-Honest-Product.pdf>
- [7] GS1, “EAN UCC Traceability Implementation”, February 2003. URL:  
<https://www.yumpu.com/en/document/read/22103258/eanoucc-traceability-implementation-gs1/85>
- [8] Accenture - CHRISTINE LEONG, TAL VISKIN, ROBYN STEWART, “TRACING THE SUPPLY CHAIN”, 2018. URL:  
[https://www.accenture.com/\\_acnmedia/PDF-93/Accenture-Tracing-Supply-Chain-Blockchain-Study-PoV.pdf](https://www.accenture.com/_acnmedia/PDF-93/Accenture-Tracing-Supply-Chain-Blockchain-Study-PoV.pdf)

- [9] T. Moe, “Perspectives on traceability in food manufacture”, Danish Institute for Fisheries Research, Trends in Food Science & Technology 9 (1998). URL:  
[http://depa.fquim.unam.mx/amyd/archivero/trazabilidad\\_1904.pdf](http://depa.fquim.unam.mx/amyd/archivero/trazabilidad_1904.pdf)
- [10] European Commission, “Farm to Fork Strategy – for a fair, healthy and environmentally-friendly food system”, Agroecology Knowledge Hub, 2020 URL:  
[https://ec.europa.eu/food/sites/food/files/safety/docs/f2f\\_action-plan\\_2020\\_strategy-info\\_en.pdf](https://ec.europa.eu/food/sites/food/files/safety/docs/f2f_action-plan_2020_strategy-info_en.pdf)
- [11] Deloitte, “Deloitte 2020 Global Blockchain Survey”, Deloitte Insights, 2020. URL:  
[https://www2.deloitte.com/content/dam/insights/us/articles/6608\\_2020-global-blockchain-survey/DI\\_CIR%202020%20global%20blockchain%20survey.pdf](https://www2.deloitte.com/content/dam/insights/us/articles/6608_2020-global-blockchain-survey/DI_CIR%202020%20global%20blockchain%20survey.pdf)
- [12] Michael Crosby, Nachiappan, Pradhan Pattanayak, Sanjeev Verma, Vignesh Kalyanaraman, “BlockChain Technology”, Sutardja Center for Entrepreneurship & Technology Technical Report, October 16, 2015. URL:  
<http://scet.berkeley.edu/wp-content/uploads/BlockchainPaper.pdf>
- [13] Stuart Haber, W. Scott Stornetta, “How to Time-Stamp a Digital Document”, Journal of Cryptology: the Journal of the International Association for Cryptologic Research 3(2), pp. 99–112, 1991. URL: <https://link.springer.com/content/pdf/10.1007/BF00196791.pdf>
- [14] Satoshi Nakamoto, “Bitcoin: A Peer-to-Peer Electronic Cash System”, 2008. URL:  
<https://bitcoin.org/bitcoin.pdf>
- [15] Vitalik Buterin, “A NEXT GENERATION SMART CONTRACT & DECENTRALIZED APPLICATION PLATFORM”, April 2014. URL:  
<https://ethereum.org/en/whitepaper/>
- [16] Sobti, Rajeev & Ganesan, Geetha, “Cryptographic Hash Functions: A Review. “, International Journal of Computer Science Issues, ISSN (Online): 1694-0814. Vol 9. 461 - 479. URL:  
[https://www.researchgate.net/publication/267422045\\_Cryptographic\\_Hash\\_Functions\\_A\\_Review](https://www.researchgate.net/publication/267422045_Cryptographic_Hash_Functions_A_Review)
- [17] Zheng Zibin, Xie Shaoan, Dai Hong-Ning, Chen Xiangping, Wang Huaimin, “An Overview of Blockchain Technology: Architecture, Consensus, and Future Trends.”, 10.1109/BigDataCongress.2017.85, 2017. URL:  
[https://www.researchgate.net/publication/318131748\\_An\\_Overview\\_of\\_Blockchain\\_Technology\\_Architecture\\_Consensus\\_and\\_Future\\_Trends](https://www.researchgate.net/publication/318131748_An_Overview_of_Blockchain_Technology_Architecture_Consensus_and_Future_Trends)
- [18] Dindayal Mahto, Dilip Kumar Yadav, “RSA and ECC: A Comparative Analysis”, International Journal of Applied Engineering Research ISSN 0973-4562 Volume 12, Number 19 (2017) pp. 9053-9061. URL: [https://www.ripublication.com/ijaer17/ijaer12n19\\_140.pdf](https://www.ripublication.com/ijaer17/ijaer12n19_140.pdf)

- [19] Demiro Massessi, “Blockchain Public / Private Key Cryptography In A Nutshell”, Oct 15, 2018. URL:  
<https://medium.com/coinmonks/blockchain-public-private-key-cryptography-in-a-nutshell-b7776e475e7c>
- [20] Lamport, L., R. Shostak and M. C. Pease. “The Byzantine Generals Problem.” ACM Trans. Program. Lang. Syst. 4 (1982): 382-401. URL:  
<https://www.microsoft.com/en-us/research/uploads/prod/2016/12/The-Byzantine-Generals-Problem.pdf>
- [21] Pavel Vasin, “BlackCoin’s Proof-of-Stake Protocol v2”, 2014. URL:  
<https://blackcoin.org/blackcoin-pos-protocol-v2-whitepaper.pdf>
- [22] King S, Scott Nadal. “PPCoin: Peer-to-Peer Crypto-Currency with Proof-of-Stake.” (2012). URL: <https://decred.org/research/king2012.pdf>
- [23] Ethereum Staking. URL: <https://ethereum.org/en/eth2/staking/>
- [24] Figueiredo Do Nascimento, S., Roque Mendes Polvora, A. and Sousa Lourenco, J., “#Blockchain4EU: Blockchain for Industrial Transformations”, EUR 29215 EN, Publications Office of the European Union, Luxembourg, 2018. URL:  
<https://publications.jrc.ec.europa.eu/repository/handle/JRC111095>
- [25] Provenance website. URL: <https://www.provenance.org/>
- [26] Mika Lammi, “SMARTLOG PILOTING BLOCKCHAIN FOR LOGISTICS”, 2017. URL: [https://www.porttechnology.org/wp-content/uploads/2019/05/036-038\\_3.pdf](https://www.porttechnology.org/wp-content/uploads/2019/05/036-038_3.pdf)
- [27] EnergyWeb. URL: <https://www.energyweb.org/>
- [28] Storj - Decentralized Cloud Storage. URL: <https://www.storj.io/>
- [29] Decentraland. URL: <https://decentraland.org/>
- [30] Telia company, ChromaWay and Kairos Future, “The Land Registry in the blockchain - testbed”, March 2017. URL:  
[https://static1.squarespace.com/static/5e26f18cd5824c7138a9118b/t/5e3c35451c2ccb6170caa19e/1581004119677/Blockchain\\_Landregistry\\_Report\\_2017.pdf](https://static1.squarespace.com/static/5e26f18cd5824c7138a9118b/t/5e3c35451c2ccb6170caa19e/1581004119677/Blockchain_Landregistry_Report_2017.pdf)
- [31] Takenobu T., “Ethereum EVM illustrated”, Rev. 0.01.1. URL:  
[https://takenobu-hs.github.io/downloads/ethereum\\_evm\\_illustrated.pdf](https://takenobu-hs.github.io/downloads/ethereum_evm_illustrated.pdf)
- [32] Jake Frankenfield, “Smart Contracts”, May 26 2021. URL:  
<https://www.investopedia.com/terms/s/smart-contracts.asp>
- [33] Wood, Gavin. "Ethereum: A secure decentralised generalised transaction ledger." Ethereum project yellow paper 151, no. 2014. URL:  
<https://ethereum.github.io/yellowpaper/paper.pdf>
- [34] OriginTrail - making supply chains work together. URL: <https://origintrail.io/>

- [35] ShipChain - The end-to-end logistics platform of the future. URL: <https://shipchain.io/>
- [36] Hyperledger - Introduction. URL:  
<https://hyperledger-fabric.readthedocs.io/en/latest/whatis.html>
- [37] Truffle Suite - Ganache. URL: <https://www.trufflesuite.com/docs/ganache/overview>
- [38] Truffle Suite. URL: <https://www.trufflesuite.com/docs/truffle/overview>
- [39] Web3.js documentation. URL: <https://web3js.readthedocs.io/>
- [40] Infura - The World's Most Powerful Blockchain Development Suite. URL:  
<https://infura.io/>
- [41] Torus - authentication and key management products. URL:  
<https://docs.tor.us/customauth/get-started>
- [42] Metamask - A crypto wallet & gateway to blockchain apps. URL: <https://metamask.io/>
- [43] Dasaklis Thomas, Casino Fran, Patsakis Costas, Douligeris Christos, “A framework for supply chain traceability based on blockchain tokens.”, 2019. URL:  
[https://www.researchgate.net/publication/335061542\\_A\\_framework\\_for\\_supply\\_chain\\_traceability\\_based\\_on\\_blockchain\\_tokens](https://www.researchgate.net/publication/335061542_A_framework_for_supply_chain_traceability_based_on_blockchain_tokens)
- [44] Torus Labs, “What Distributed Key Generation Is”, Feb 15, 2020 .URL:  
<https://medium.com/toruslabs/what-distributed-key-generation-is-866adc79620>
- [45] Pieter Wuille, “Hierarchical Deterministic Wallets”, 2012-02-11. URL:  
<https://github.com/bitcoin/bips/blob/master/bip-0032.mediawiki>
- [46] Everledger - Tech for Good Blockchain Solutions. URL: <https://www.everledger.io/>
- [47] Javascript QR Code Scanner. URL: <https://github.com/nimiq/qr-scanner>

# Anexa 1 - Arhitectura Blockchain

Blockchain este o bază de date distribuite ce conține un registru (ledger) deschis al tuturor tranzacțiilor sau evenimentelor ce au fost partajate între entitățile participante [12]. Fiecare tranzacție din acest registru este verificată printr-un algoritm ce determină consensul majorității.

## 1.1 Iстория

Principiile de funcționare ale tehnologiei blockchain au fost definite în anul 1991 de Stuart Haber și Scott Stornetta în lucrarea "How To Time-Stamp a Digital Document" [13], prin care aceștia au dorit să implementeze un sistem prin care data de semnare a unui document să nu poată fi schimbată.

După aproape două decenii, prima aplicare reală a venit odată cu inventia protocolului de plată Bitcoin, lansat în anul 2009 de entitatea pseudo anonima Satoshi Nakamoto [14]. Bitcoin reprezintă prima monedă virtuală descentralizată (criptomonedă).

Incepând din 2013 au apărut noi aplicații ale tehnologiei, precum platforma Ethereum [15], cunoscute generic sub numele de blockchain 2.0. Acestea au introdus în principal conceptul de contracte inteligente, ce reprezintă un mod de a rula segmente de cod pe blockchain pentru a efectua diferite operații.

## 1.2 Arhitectura

Datele din blockchain, denumit și lant de blocuri, sunt împărțite în înregistrări numite blocuri conectate precum o listă simplu înlanțuită. Fiecare bloc conține un hash criptografic al blocului anterior, o marca temporală (timestamp), și datele tranzacției. În Figura 69 se poate vedea o ilustrare a unui blockchain simplificat.

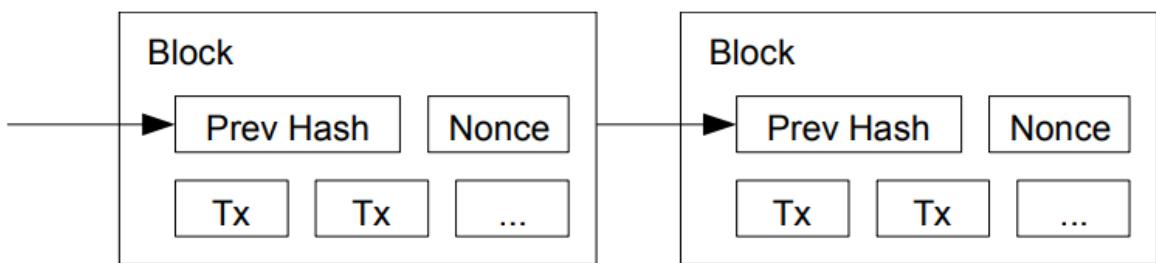


Figura 69: Înlănțuirea blocurilor [14]

## Funcții hash

Un hash este generat de o funcție hash (funcție de dispersie). Acestea sunt funcții definite pe o mulțime cu o infinitate de elemente, cu valori dintr-o mulțime cu un număr finit de elemente. Cateva dintre proprietățile esențiale pe care o funcție hash trebuie să le satisfacă sunt [16]:

- Reversibilitatea. Dat rezultatul unei funcții hash, nu este posibil să ajungem la rezultatul inițial (One way function)
- Efectul de avalanșă și completitudinea. Toți biții mesajului inițial sunt folosiți pentru a genera rezultatul, iar modificarea unuia dintre ei la intrare duce la o schimbare semnificativă al celor rezultați.
- Rezistența la coliziuni. Trebuie să fie infailibil să găsim 2 mesaje care să producă același hash.
- Determinism. Funcția trebuie să genereze același rezultat pentru aceleasi date de intrare.
- Opacitate. Rezultatul nu trebuie să conțină indicații cu privire la datele de intrare.

Hash-ul fiecărui bloc este format din datele sale plus hash-ul blocului precedent, astfel creându-se un lanț de blocuri legate din punct de vedere criptografic. Prin aceasta abordare, dacă va exista o modificare în unul din blocurile anterioare, valorile hash vor trebui modificate pe tot traseul până la blocul curent.

## Structura unui bloc

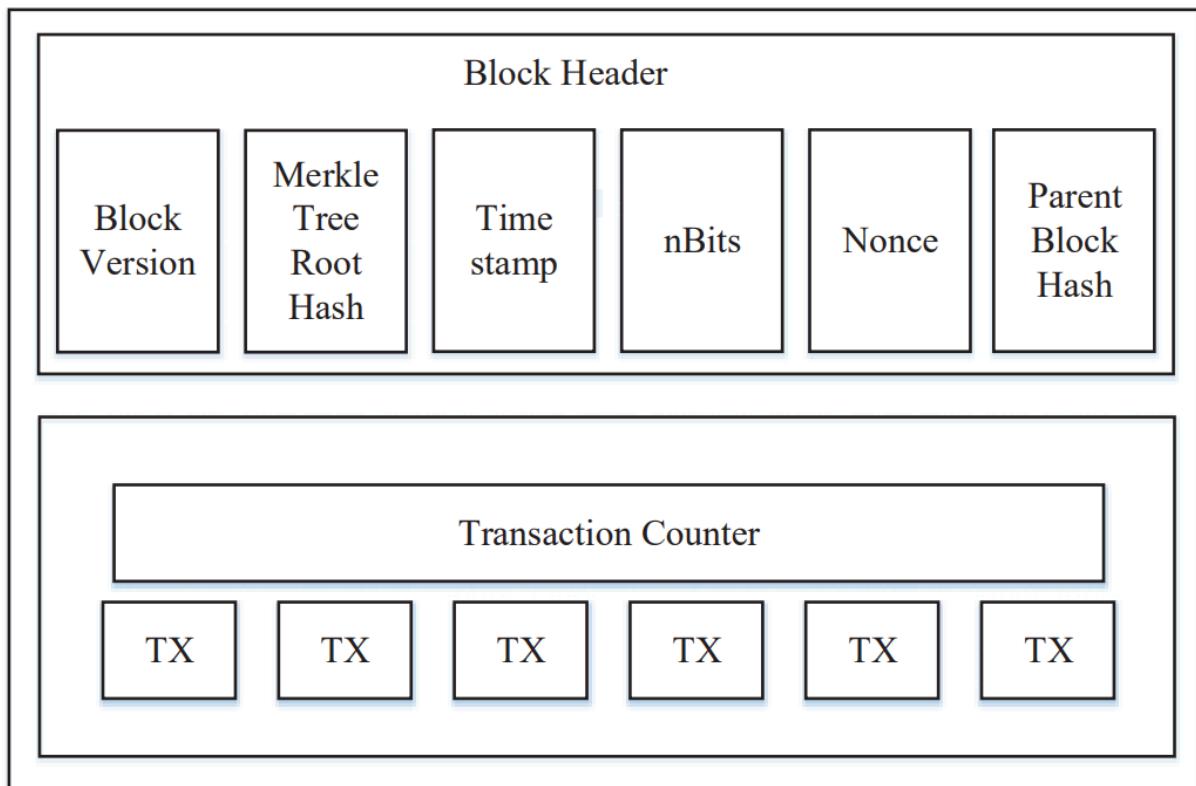


Figura 70: Structura unui bloc [17]

Precum în Figura 70, datele unui bloc sunt organizate în două categorii: capul blocului (block header) și corpul blocului (block body). Capul blocului conține în principal următoarele informații:

- Block version - indicator asupra modului în care se realizează validarea acestuia
- Merkle tree root hash - hash-ul radacini arborelui Merkle. Arborele Merkle este o structură de date arborescentă în care fiecare nod frunză este etichetat cu hash-ul unor date specifice aplicației, iar nodurile non frunză sunt etichetate cu hash-urile nodurilor copil. În contextul unui blockchain, acest lucru este util pentru a putea verifica o tranzacție într-un mod rapid și economic, fără a descărca tot conținutul blocurilor anterioare, precum în Figura 72.

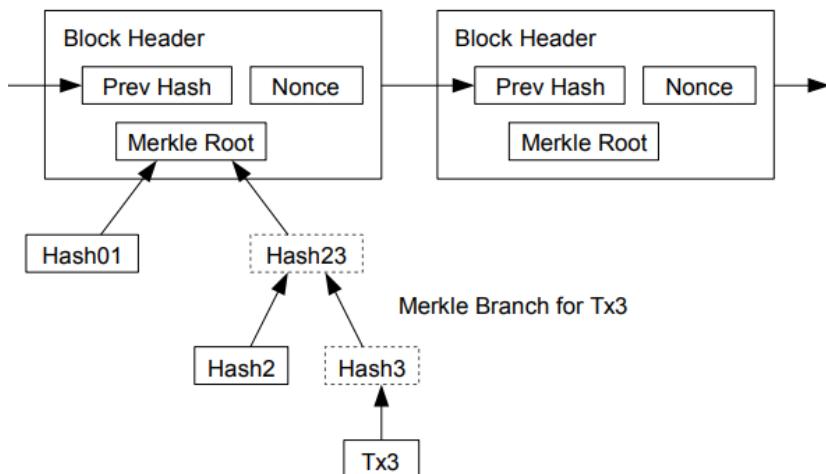


Figura 72: Arbore Merkle în contextul unui blockchain [14]

- Timestamp - data și ora la care a fost adăugat blocul
- nBits - numărul de biți al unui număr sub care un hash valid al blocului curent trebuie găsit
- Nonce - Un număr unic al blocului. Aceasta este folosit pentru a calcula diferite hash-uri
- Parent block hash - Valoarea hash al blocului părinte

Corpul blocului este alcătuit dintr-o listă de tranzacții și un contor al acestora. Numărul maxim de tranzacții depinde de dimensiunea maximă stabilită al unui bloc și de dimensiunea unei tranzacții.

## Semnături digitale

În paradigma blockchain, entitățile care efectuează tranzacții, sunt reprezentate printr-o pereche de chei, o cheie privată și o cheie publică. Conform modului de lucru al criptografiei asimetrice, cheia privată este confidentială și este utilizată pentru a semna o tranzacție. După semnare, tranzacția este trimisă în rețea către un nod, iar acesta o trimite mai departe tuturor nodurilor conectate.

Un algoritm des folosit este ECDSA (elliptic curve digital signature algorithm). Acesta aduce beneficii substanțiale din punct de vedere al performanței fata de RSA [18].

În Figura 73, este exemplificat cum o tranzacție este semnată cu cheia privată, iar apoi aceasta poate fi verificată din punct de vedere al validității cu cheia publică.

## Digital Signature

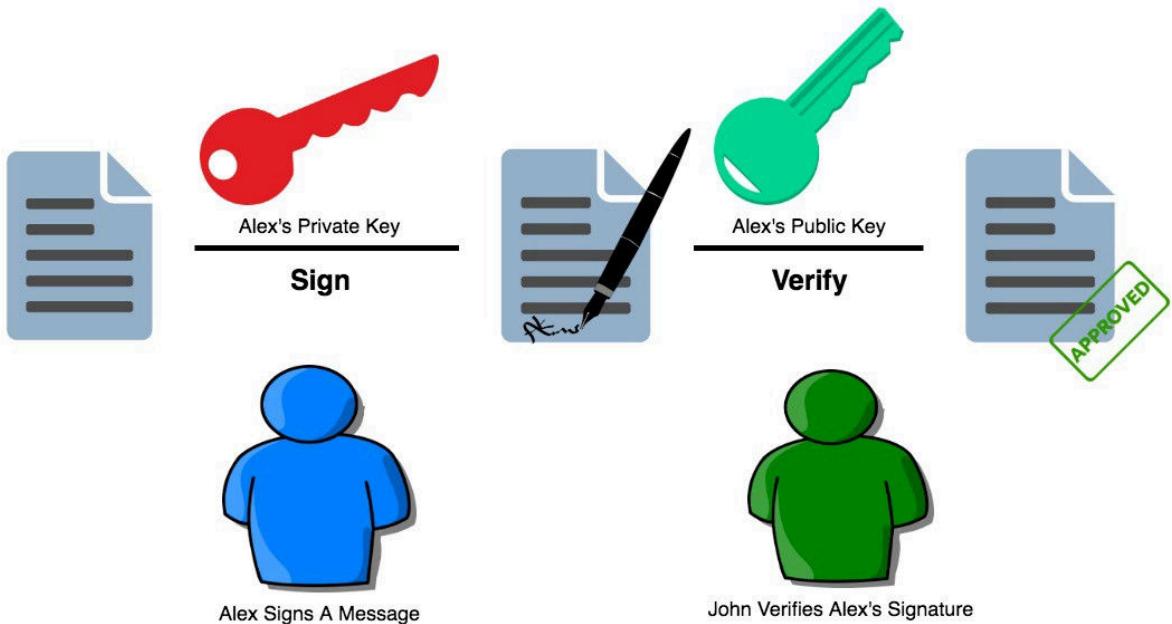


Figura 73: Semnarea unei tranzacții [19]

### Algoritmi de consens

Deși prin modul de înlățuire al blocurilor, este garantată securitatea din punct de vedere al modificării datelor deja existente, natura descentralizată implica adaugarea de noi blocuri prin intermediul unor noduri potențial nesigure conectate la rețea. Aceasta problema este cunoscută în domeniu sub numele de “Problema generalilor bizantini” - “Byzantine Generals (BG) Problem”, ce a fost definită prima dată în anul 1982 [20]. Pe scurt aceasta prezintă problema unor generali bizantini ce au probleme în a se hotărî în același timp, dacă ataca sau se retrag. Proprietatea BFT - Byzantine Fault Tolerance este proprietatea unui sistem de a continua să opereze chiar dacă unele entități ale acestuia sunt răuvoitoare sau dispar.

Un algoritm de consens este un mecanism prin care un sistem distribuit, o rețea blockchain, ajunge la un consens asupra blocurilor și tranzacțiilor ce vor fi adăugate. Cei mai cunoscuți algoritmi de consens sunt Proof of Work (PoW) și Proof of Stake (PoS).

**Proof of Work** se bazează pe puterea de calcul al unui nod. Modul de operare al acestui algoritm este cunoscut sub numele de mining, și presupune ca pentru fiecare bloc ce va fi adăugat, nodurile din rețea, minerii, trebuie să găsească o valoare al parametrului nonce din bloc astfel încât hash-ul blocului să fie mai mic sau egal decât o valoare specificată

de parametrul nBits. Cand unul din noduri găsește valoarea nonce care să satisfacă condiția, acesta o adaugă în block și îl trimită către ceilalți participanți ai rețelei care vor verifica corectitudinea calcului. În cazul în care 2 sau mai multe blocuri sunt adăugate în același timp, precum în Figura 74, blockchain-ul se împarte în două sau mai multe bifurcații, iar în momentul în care una din ele devine mai lungă decât celelalte, aceasta este aleasă ca fiind cea autentică. De exemplu, în cazul Bitcoin, dificultatea adică valoarea sub care hash-ul unui bloc trebuie găsit, este ajustată în mod periodic și automat în funcție de puterea de calcul a nodurilor, astfel încât un bloc să fie adăugat la fiecare 10 minute. O vulnerabilitate adesea discutată este atacul 51% (51% Attack) care se referă la posibilitatea ca un grup de mineri să controleze mai mult de 50% din puterea de procesare a întregii rețele astfel încât să poată bloca sau trimite de mai multe ori anumite tranzacții. În teorie, o formă al acestui atac este posibilă și cu mai puțin de 50% din puterea de calcul a rețelei, dar cu o probabilitate mai mică de reușita.

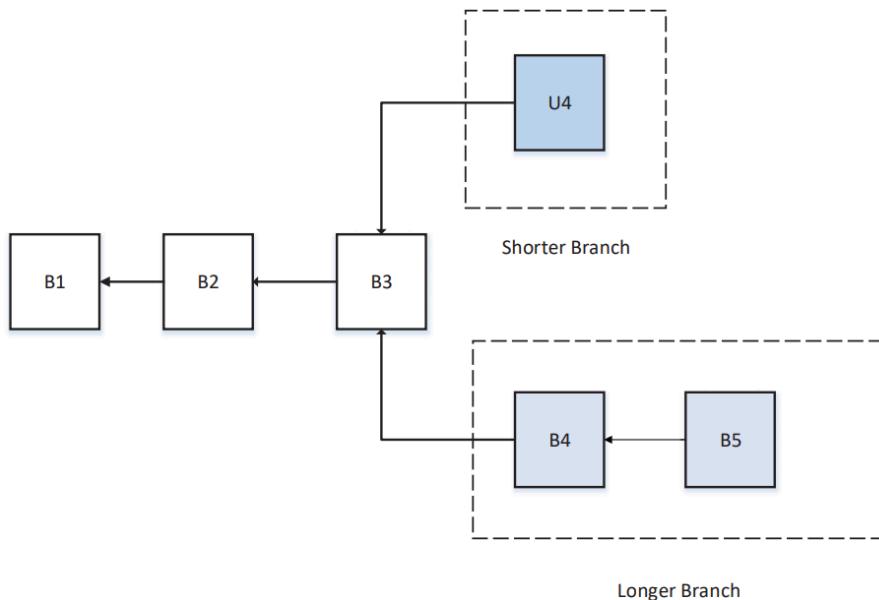


Figura 74: Cea mai lungă bifurcație este aleasă [17]

**Proof of Stake** este o alternativă mai economică din punct de vedere al necesității computaționale. Nodurile trebuie să dovedească încrederea prin deținerea unui activ valoros din punct de vedere financiar, acesta fiind de obicei moneda implementată în respectivul blockchain. Deși se consideră că nodurile cu cele mai mari resurse financiare sunt cele mai de încredere, acest lucru poate duce la monopolizarea rețelei de către jucători principali. Pentru a se remedia aceasta problema, diferite euristică au fost implementate precum randomizarea nodului ce va fi validator [21] sau favorizarea nodurilor ce au o vechime mai mare [22]. Din cauza consumului mare de energie al algoritmului Proof of Work, multe dintre blockchain-uri au început adaptarea acestora către Proof of Stake, printre acestea se numără și platforma Ethereum [23].

### 1.3 Tipuri de blockchain

Blockchain-urile pot fi de mai multe tipuri:

- Public - oricine are acces la internet poate participa în procesul de trimisare, vizualizare și verificare ale tranzacțiilor.
- Privat - un blockchain privat este unul centralizat, unde doar o singură entitate are acces.
- Consortiu - similar cu un blockchain privat, dar controlat de mai multe organizații astfel, cu un nivel de descentralizare mai mare
- Hibrid - acesta este format dintr-o combinație din două parti, una privată și una publică

### 1.4 Utilizari în industrii

Aplicabilitatea acestei tehnologii este extrem de vastă, cîteva dintre posibilele arii sunt votul electronic, finanțe descentralizate (DeFi), aplicații descentralizate (DApps), drepturi de autor, colecționarea de bunuri digitale (NFT), organizații autonome descentralizate (DAO) etc.

Tehnologia blockchain este considerată ca va avea o contribuție deosebit de importantă în viitoarea transformare a organizațiilor, guvernarea democratică și cultura umană în ansamblu. Proiectul de studiu "Blockchain4EU" realizat de Uniunea Europeană își propune să analizeze potențialul tehnologiei blockchain în domeniul industrial și să faciliteze adoptarea unor politici potrivite noilor schimbări tehnologice [24]. Cîteva dintre sectoarele industriale în care blockchain are un impact major sunt:

- Producerea și distribuirea alimentelor - Startup-ul Provenance[25] lucrează cu peste 200 de producători și distribuitori alimentari pentru a facilita transparentă din punct de vedere al provenientei și calităților fizice ale produselor
- Transport și Logistică - Proiectul SmartLog [26] este o soluție bazată pe blockchain care are ca scop transferul de date cu usurință între entitățile logistice pentru a reduce timpul de transport și a monitoriza transporturile.
- Sanitate și industria farmaceutică - Blockchain poate avea un impact în transparentizarea datelor cu privire la studiile clinice ale medicamentelor
- Domeniul creativ - Cu ajutorul conceptului de NFT - Non Fungible Token oricine își poate pune la vânzare diferite active digitale. Un exemplu de platformă de tranzacționare a NFT-urilor este Decentraland [29].

- Energie - Energy Web Foundation [27] dezvolta un blockchain conectat la infrastructura energetica existentă pentru a satisface diferite nevoi operaționale ale pieței.
- Tehnologia informației - Proiectul Storj[28] oferă o soluție descentralizată și criptare end-to-end de stocare a fișierelor, iar participantii la rețea își pot închiria spațiul de stocare în schimbul unei remuneratii financiare.
- Resurse naturale - Compania ChromaWay împreuna cu un stat din India au realizat un proiect de cercetare prin care tehnologia blockchain este folosită pentru a facilita vanzarea și cumpararea de terenuri [30].

## Anexa 2 - Platforma Ethereum

### 2.1 Introducere

Ethereum a fost lansat la sfârșitul anului 2013 de Vitalik Buterin împreună cu Mihai Alisie, Anthony Di Iorio și Charles Hoskinson [15].

### 2.2 Arhitectura

Ethereum este o platformă pentru a dezvolta aplicații descentralizate, bazată pe un blockchain programabil care rulează pe o mașină virtuală - Ethereum virtual machine (EVM). Principiul de bază în ecosistemul ethereum este că există un singur computer global al căruia stare este modificată de fiecare tranzacție și este salvată în blockchain, precum în Figura 75 și Figura 76.

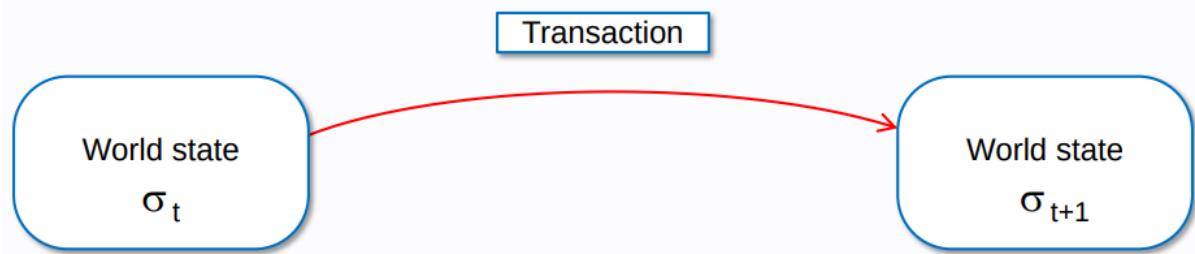


Figura 75: O transacție reprezintă un arc valid între două stări [31]

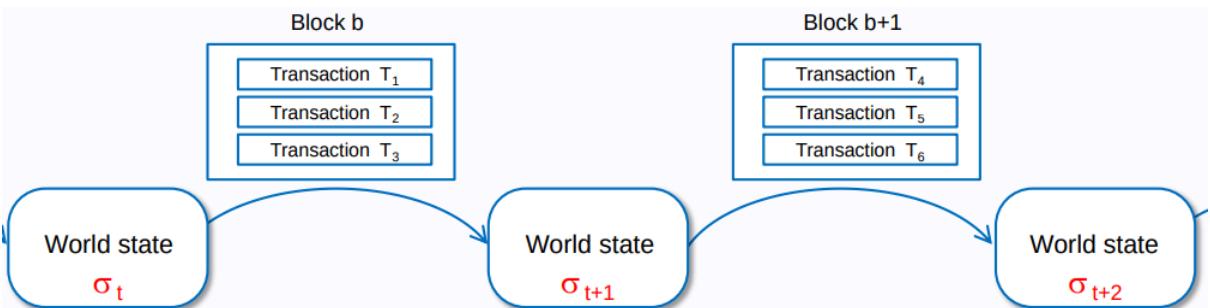


Figura 76: Tranzacțiile sunt stocate în blocuri. Ethereum poate fi considerat un lanț de stări. [31]

Fiecare nod din rețea are o copie a blockchain-ului și rulează o instanță EVM.

### Ether

Ether (ETH), moneda virtuală nativă a platformei Ethereum, are ca scop principal remunerarea din punct de vedere financiar al nodurilor din rețea pentru verificare și executarea tranzacțiilor. Pe lângă faptul că moneda se poate trimite între conturi, aceasta este utilizată pentru a plăti munca computatională a unui nod. Atunci când un utilizator dorește să efectueze o tranzacție, acesta trebuie să specifică și o cantitate de ether, care va fi mai apoi

trimisă către nodul care va verifica, executa, salva și trimite blocul cu tranzacția către celelalte noduri.

## Contracte inteligente - Smart Contracts

Nick Szabo, un informatician american care a inventat o monedă virtuală numită „Bit Gold” în 1998, a definit contractele inteligente drept protocoale de tranzacții computerizate care execută termenii unui contract [32].

Un contract intelligent - smart contract, este un fragment reutilizabil de cod pe care un programator îl publică pe blockchain. Orice utilizator poate solicita execuția respectivului contract printr-o tranzacție. În general aplicațiile care sunt bazate pe smart contracts se numesc aplicații descentralizate sau DApps. Contractele inteligente pot fi scrise în diferite limbaje de programare precum Solidity, Vyper, Bamboo, Serpent, Mutan, etc, cel mai folosit în momentul actual fiind Solidity. În Figura 77 putem vedea un exemplu de un contract intelligent.

```
pragma solidity ^0.4.0;

contract SimpleStorage {
    uint storedData;

    function set(uint x) public {
        storedData = x;
    }

    function get() public view returns (uint) {
        return storedData;
    }
}
```

Figura 77: Un exemplu de contract intelligent

Contractele sunt compilate în instrucțiuni low-level, numite opcodes, care sunt apoi rulate de EVM, mașina virtuală Ethereum. Acestea permit EVM să obțină proprietatea de Completitudine Turing (Turing completeness) ceea ce înseamnă că un program realizat poate rezolva orice problema computatională (fără vreo garanție al timpului de execuției sau memoriei utilizate). În Figura 78 sunt prezentate câteva dintre codurile de operații executate

de EVM, după compilarea contractului.

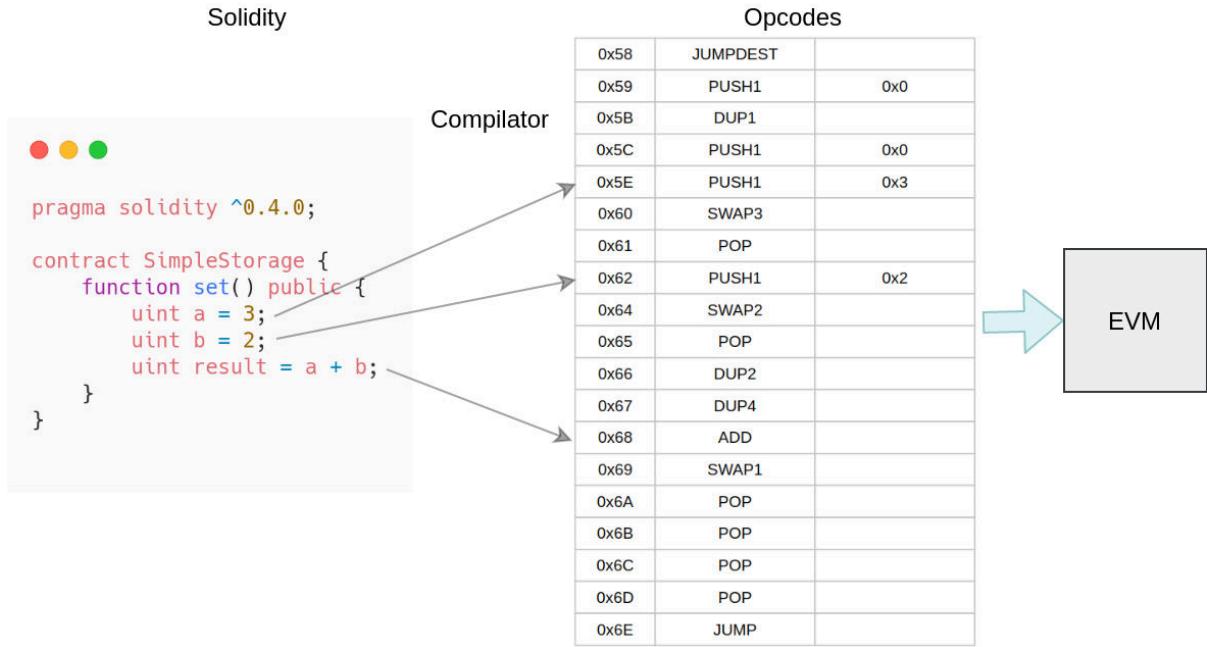


Figura 78: De la codul Solidity la opcodes

Deoarece execuția unui contract este realizată pe toate nodurile rețelei, un atacator ar putea crea un contract continând bucle infinite sau operații extrem de costisitoare pentru a suprasolicita rețea. Pentru prevenirea acestui lucru, fiecare opcode are un cost numit gas (benzina). De exemplu, pentru a realiza operația de adunare trebuie să folosim instrucțiunea ADD, iar conform figurilor 79 și 80, ADD se află în multimea  $W_{\text{verylow}}$  cu un cost asociat de 3 gas. Astfel, în cazul existenței unei bucle infinite, EVM va rula un număr limitat de iterări. Pe lângă costurile fixe, există instrucțiuni ce au costuri dinamice precum KECCAK256 (similar cu SHA3) care au un cost de bază de 30 gas și un cost dinamic de 6 gas pe cuvânt (un cuvânt înseamnă un sir de caractere de 256 de biți).

#### APPENDIX G. FEE SCHEDULE

The fee schedule  $G$  is a tuple of 31 scalar values corresponding to the relative costs, in gas, of a number of abstract operations that a transaction may effect.

Name	Value	Description*
$G_{\text{zero}}$	0	Nothing paid for operations of the set $W_{\text{zero}}$ .
$G_{\text{base}}$	2	Amount of gas to pay for operations of the set $W_{\text{base}}$ .
$G_{\text{verylow}}$	3	Amount of gas to pay for operations of the set $W_{\text{verylow}}$ .
$G_{\text{low}}$	5	Amount of gas to pay for operations of the set $W_{\text{low}}$ .
$G_{\text{mid}}$	8	Amount of gas to pay for operations of the set $W_{\text{mid}}$ .
$G_{\text{high}}$	10	Amount of gas to pay for operations of the set $W_{\text{high}}$ .
$G_{\text{extcode}}$	700	Amount of gas to pay for an EXTCODESIZE operation.
$G_{\text{extcodehash}}$	700	Amount of gas to pay for an EXTCODEHASH operation.
$G_{\text{balance}}$	700	Amount of gas to pay for a BALANCE operation.
$G_{\text{sload}}$	800	Paid for a SLOAD operation.
$G_{\text{jumpdest}}$	1	Paid for a JUMPDEST operation.
$G_{\text{set}}$	20000	Paid for an SSTORE operation when the storage value is set to non-zero from zero.
$G_{\text{reset}}$	5000	Paid for an SSTORE operation when the storage value's zeroness remains unchanged or is set to zero.
$R_{\text{sclear}}$	15000	Refund given (added into refund counter) when the storage value is set to zero from non-zero.
$R_{\text{selfdestruct}}$	24000	Refund given (added into refund counter) for self-destructing an account.
$G_{\text{selfdestruct}}$	5000	Amount of gas to pay for a SELFDESTRUCT operation.

Figura 79: Costurile unor tipuri de operații [33]

```

 $W_{zero} = \{\text{STOP, RETURN, REVERT}\}$ 
 $W_{base} = \{\text{ADDRESS, ORIGIN, CALLER, CALLVALUE, CALLDATASIZE, CODESIZE, GASPRICE, COINBASE, TIMESTAMP, NUMBER, DIFFICULTY, GASLIMIT, RETURNDATASIZE, POP, PC, MSIZE, GAS}\}$ 
 $W_{verylow} = \{\text{ADD, SUB, NOT, LT, GT, SLT, SGT, EQ, ISZERO, AND, OR, XOR, BYTE, SHL, SHR, SAR, CALLDATALOAD, MLOAD, MSTORE, MSTORE8, PUSH*, DUP*, SWAP*}\}$ 
 $W_{low} = \{\text{MUL, DIV, SDIV, MOD, SMOD, SIGNEXTEND}\}$ 
 $W_{mid} = \{\text{ADDMOD, MULMOD, JUMP}\}$ 
 $W_{high} = \{\text{JUMPI}\}$ 

```

Figura 80: Operațiile din mulțimea W [33]

Inițiatorul unei tranzacții trebuie să specifică valoarea maximă al cantității de gas pe care dorește să o utilizeze execuția contractului (gas limit), cât și prețul per unitate de gas (gas price). Costul total al unei tranzacții este cantitatea de gas folosită înmulțit cu prețul per unitate de gas.

În momentul compilării, este generat și un fisier json ce conține ABI - Application binary interface, utilizat pentru a interacționa cu un contract. Fișierul conține o listă cu funcțiile contractului, mai exact cu antetul acestora, precum în Figura 81.

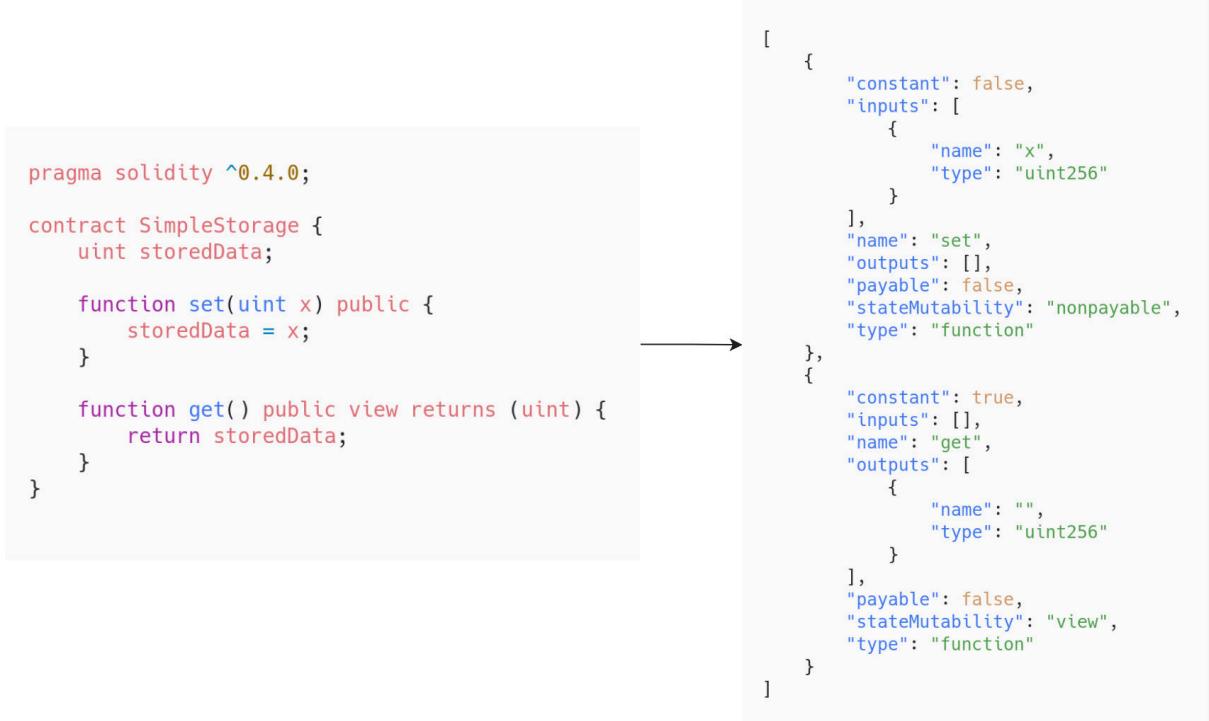


Figura 81: ABI al unui smart contract

Atunci cand dorim sa apelăm o funcție sau sa luăm anumite date dintr-un smart contract, referirea funcțiilor sau a variabilelor globale se realizeaza pe baza fișierului ABI, cu anumite procesari precum în Figura 82:

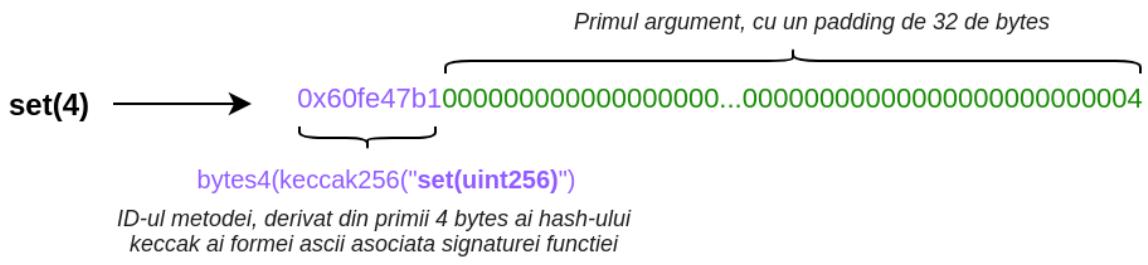


Figura 82: Modul de codare al unui apel de funcție

În arhitectura Ethereum există două moduri de interacțiune cu un contract intelligent.

**Modul Tranzacție (Transaction)** este folosit pentru a apela anumite funcții din contract care modifică starea blockchain-ului (non-view functions). Pentru ca un apel printr-o tranzacție să fie finalizat, acesta trebuie să fie trimisă către un nod și apoi acel nod va îl va trimite către toată rețeaua Ethereum, astfel tranzacția este executată și inclusă într-un bloc de fiecare nod. Datorită faptului că aceasta modifică starea, emitera unei tranzacții implica și un cost financiar (gas). Intern, transaction este realizată prin metoda RCP “eth\_sendTransaction”.

**Modul Apel (Call)** este folosit pentru a apela anumite funcții din contract ce nu modifică starea blockchain-ului (view functions). De obicei aceste funcții au rolul de a lua anumite date sau să facă procesări ce nu necesită salvarea rezultatului. Apelarea unei funcții prin modul apel se realizează doar în contextul nodului conectat. Rezultatul este întors instant, fără a mai fi nevoie de includerea acestuia într-un bloc, iar costul este 0. Intern, apelul este realizat prin metoda RCP “eth\_call”.

## Anexa 3 - Diagrame UML

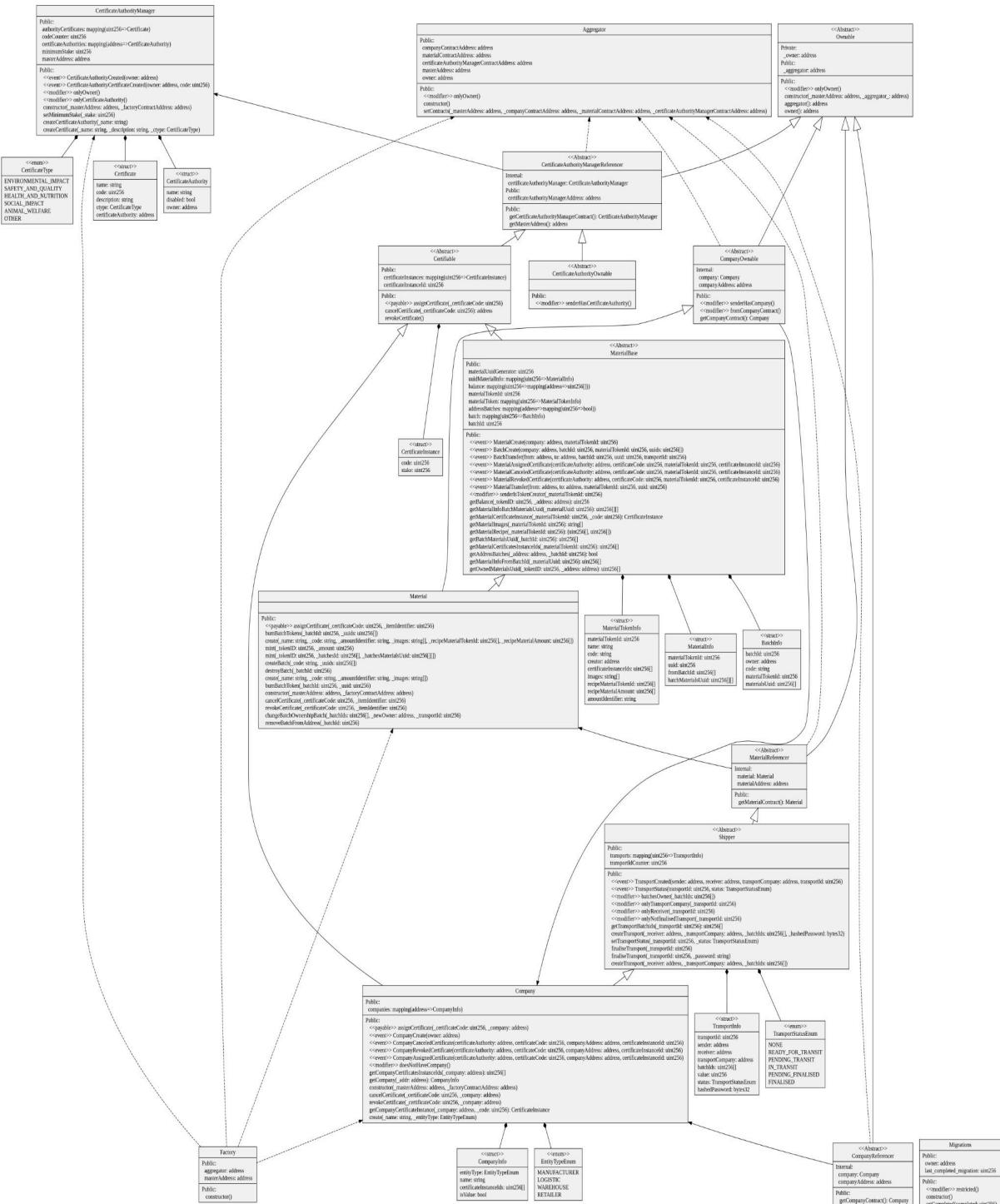


Figura 1: Diagrama UML a întregii arhitecturi de contracte inteligente

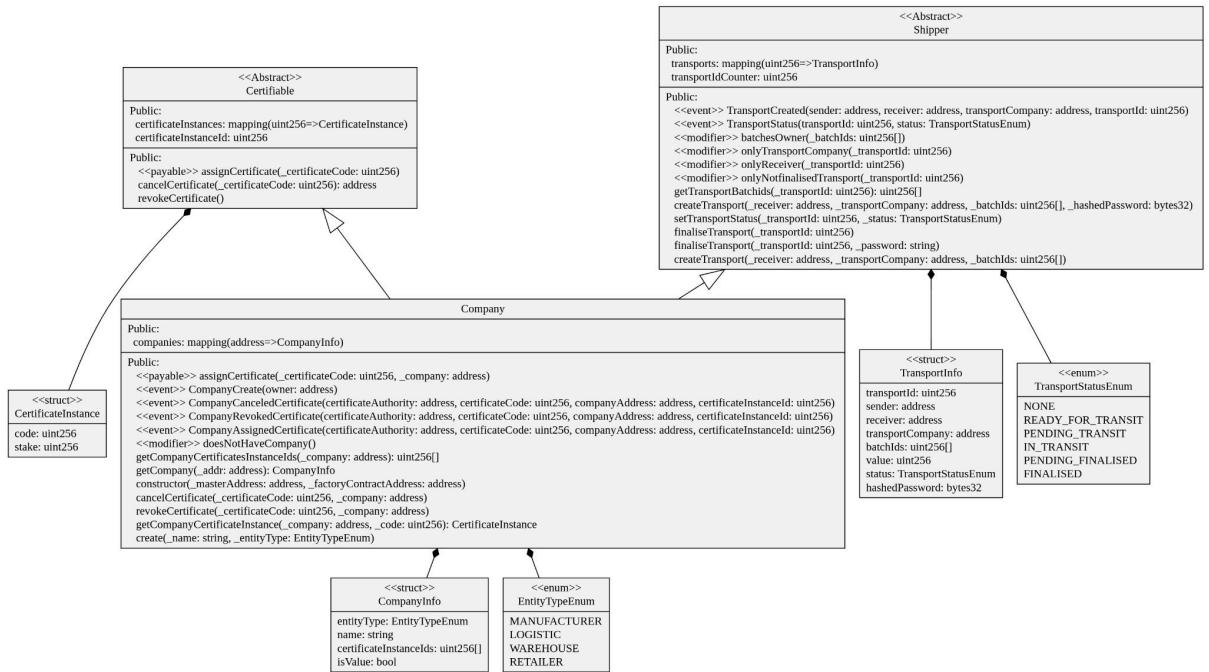


Figura 2: Diagrama UML al contractului ce administrează companiile

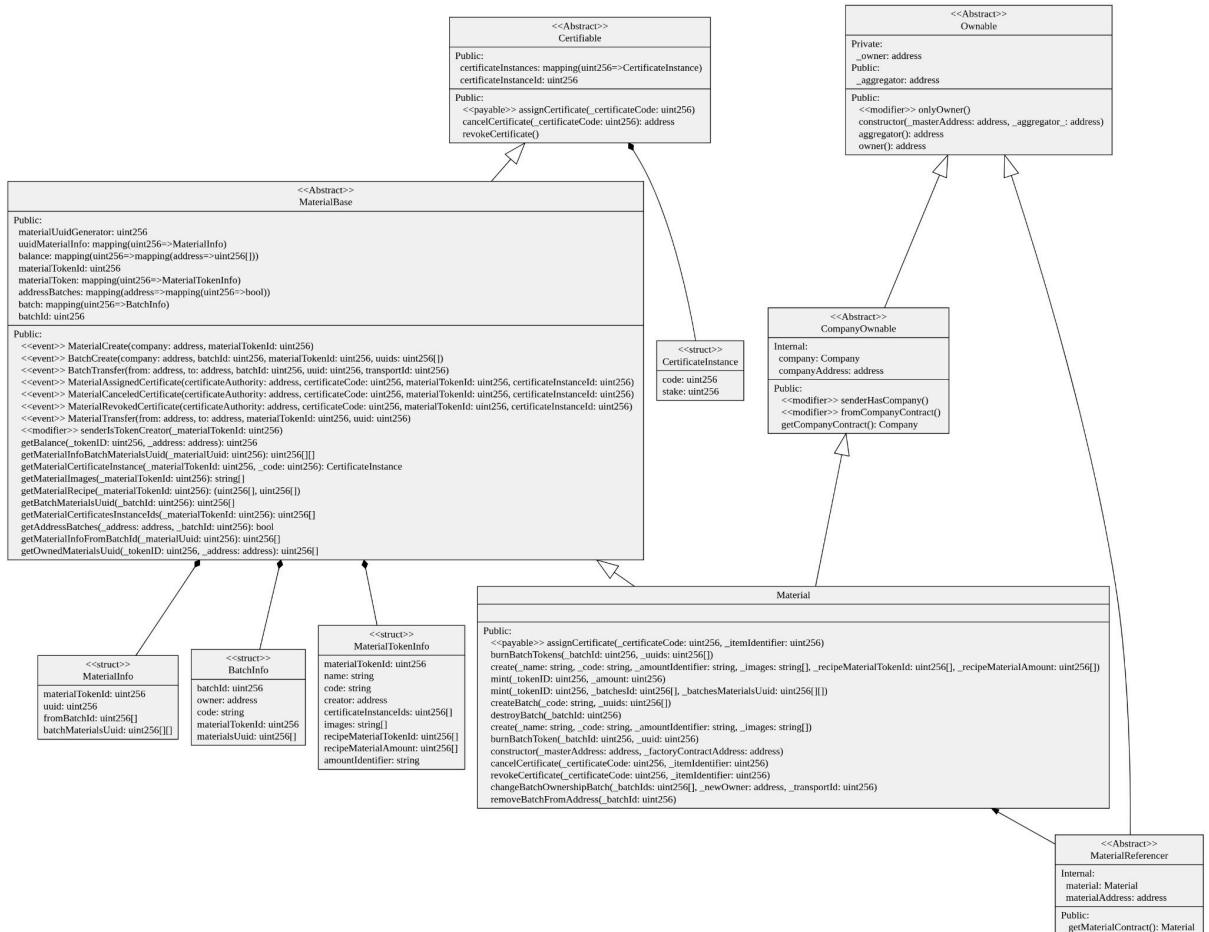


Figura 3: Diagrama UML al contractelor ce administrează materialele

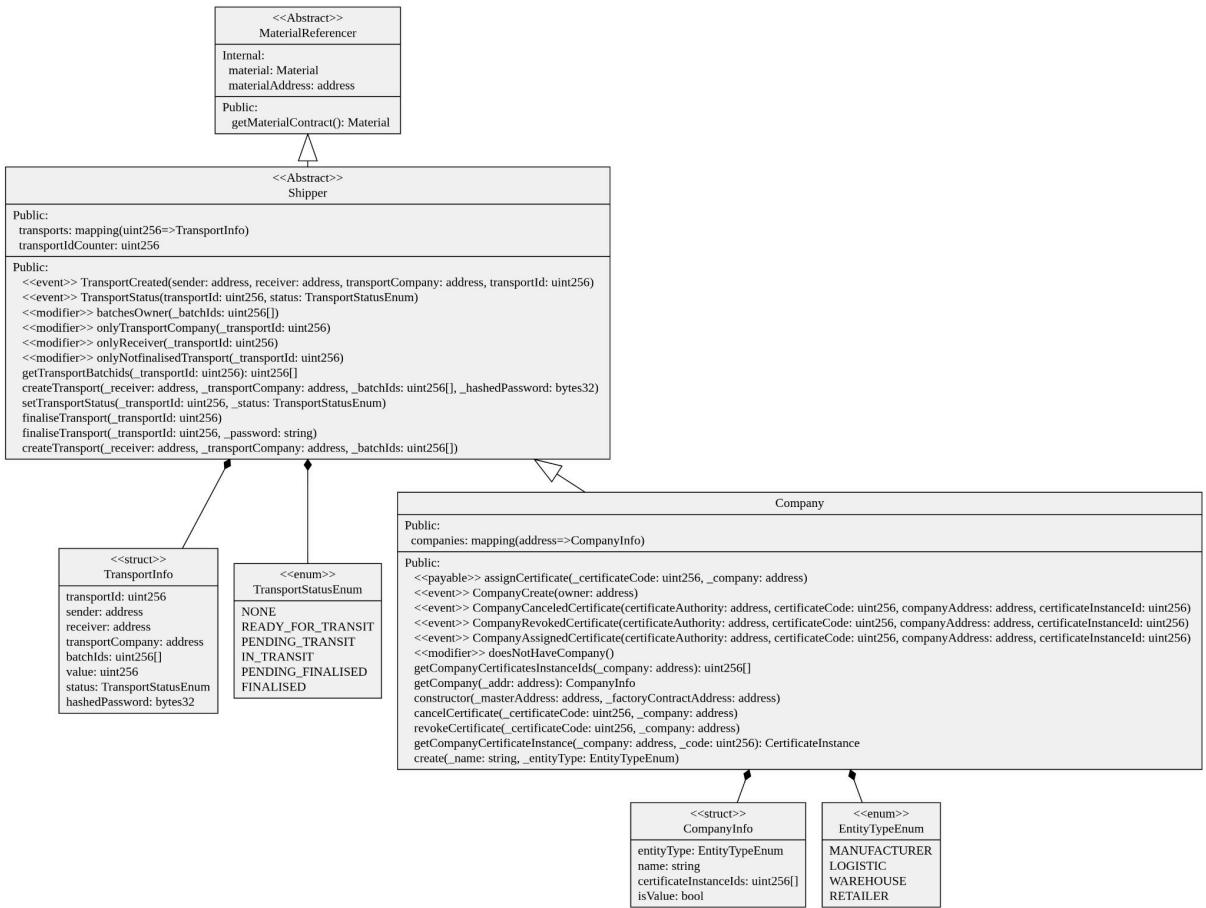


Figura 4: Diagrama UML a contractelor Shipper și Company