Politecnico di Torino

Dipartimento di Ingegneria Gestionale e della Produzione

Corso di Laurea Magistrale in Ingegneria Gestionale

Tesi di Laurea Magistrale

Tecnologia Blockchain a garanzia della trasparenza nel settore agro-alimentare:

sviluppo di un'applicazione decentralizzata basata su Ethereum per un'azienda vitivinicola.

Relatrice	Candidata
Prof.ssa Valentina GATTESCHI	Paola CIRONE

A.A 2022/2023

Abstract

Negli ultimi decenni è aumentata, in maniera significativa, la richiesta da parte dei consumatori di trasparenza sull'origine, sulla trasformazione e distribuzione dei prodotti alimentari, e, di conseguenza, la domanda di certificazioni e di sistemi di tracciabilità affidabili. Alle aziende tocca quindi ricorrere alle nuove tecnologie per far fronte a questa crescente richiesta di trasparenza. Oggi, le imprese possono perseguire questo obiettivo grazie alla tecnologia Blockchain. Nata da diversi decenni ma in continuo sviluppo, la tecnologia Blockchain, struttura dati condivisa e immutabile, è uno strumento che può consentire la tracciabilità del settore agroalimentare. Le sue caratteristiche peculiari quali decentralizzazione, trasparenza, sicurezza e immutabilità, ne consentono la piena applicazione nel settore agroalimentare, con un duplice beneficio: la Blockchain permette alle imprese di certificare la qualità e la sostenibilità dei processi produttivi, aumentando la sicurezza percepita da parte dei consumatori e, ai consumatori, di esplorare l'origine del prodotto, le tecniche colturali e le trasformazioni subite dal prodotto stesso. Il presente lavoro di tesi vuole analizzare l'uso della Blockchain nel settore agroalimentare e, dopo un'approfondita analisi dei requisiti e uno studio delle tecnologie esistenti, definire un prototipo di applicazione decentralizzata (dApp) pensata per un'azienda del settore vitivinicolo. La dApp, applicazione web il cui back-end viene eseguito su una rete decentralizzata, sfrutta gli smart contract e la Blockchain per archiviare meticolosamente documenti e dati, rendendoli consultabili da consumatori, fornitori e autorità di regolamentazione in qualsiasi momento senza poter essere manipolati. L'obiettivo è notarizzare le informazioni tramite blockchain e renderle fruibili tramite la scansione di un QR-code.

Indice

Elenco delle figure

Elenco delle tabelle

Capitolo 1

Introduzione

Il presente lavoro di tesi nasce da una certezza: il consumatore consapevole chiede trasparenza e garanzie sull'origine e lavorazione dei prodotti agroalimentari. Nel 2016 l'indagine "Global Health and Ingredient-Sentiment" condotta da NielsenIQ[1] su un campione di oltre 30000 consumatori in 61 Paesi ha evidenziato che:

- il 71% dei consumatori predilige prodotti provenienti da aziende che sono trasparenti riguardo all'origine, modalità di produzione, allevamento e coltivazione degli stessi;

- il 67% dei consumatori è preoccupato dell'uso indiscriminato di ingredienti non naturali e/o non controllati, potenzialmente dannosi sul lungo termine per la salute della persona;

- il 53%, infine, è disposto a pagare un prezzo maggiore per prodotti che non contengano ingredienti indesiderati.

[1] https://nielseniq.com/global/it/insights/analysis/2016/food-and-health-italian-ever-more-aware-and-look-to-the-ingredients/

In Italia, il sistema agro-alimentare rappresenta il 25%[2] del PIL nazionale. L'Italia si posiziona prima in Europa per valore aggiunto agricolo: la sua filiera agroalimentare vale 538 miliardi di euro, con 740mila aziende agricole, 70mila aziende alimentari, con più di 330mila attività di ristorazione e 230mila punti vendita al dettaglio che impiegano ben 4 milioni di lavoratori. Nel 2015, Giovanni Pugliese, Rappresentante Permanente Aggiunto d'Italia presso l'Unione Europea a Bruxelles, nel convegno sul tema "Come la tracciabilità può sbloccare il potenziale del settore agroalimentare in Europa", tenutosi su iniziativa dell'Enea, ha sottolineato che "*[...] per un Paese come l'Italia, la tracciabilità é un concetto che aiuta ad avvicinare il consumatore alla qualità, non solo alla sicurezza, ed é forte l'interesse dell'Italia a incoraggiare questo sistema senza eccessive semplificazioni*". Oggi, le imprese del settore agro-alimentare per perseguire e garantire al consumatore finale sicurezza e qualità dei loro prodotti possono ricorrere alla tecnologia blockchain le cui caratteristiche peculiari quali decentralizzazione, trasparenza, sicurezza e immutabilità ne consentono la piena applicazione in questo comparto. Grazie all'utilizzo della Blockchain le imprese potranno certificare la qualità e la sostenibilità dei processi produttivi; il consumatore, dal canto suo, potrà esplorare l'origine del prodotto, le tecniche colturali e le trasformazioni subite dal prodotto stesso ottenendo così le garanzie di trasparenza che oggi più che mai richiede. Attraverso innovazione e digitalizzazione si crea una nuova esperienza di acquisto, dove devono essere instaurate relazioni rassicuranti tra clienti e produttori.

[2]https://www.agrifoodtoday.it/speciale/cibus/filiera-agroalimentare-italiana-perche-e-importante.html

1.1 Il progetto

Il lavoro di tesi viene svolto in collaborazione con *Sblockchain Project*, start-up innovativa con sede legale a Torino che nasce dalla pluriennale esperienza dei promotori in vari aspetti dell'attività imprenditoriale, con un focus specifico sulle PMI (Piccole e Medie Imprese). Sblockchain Project, che rientra nel settore di servizi e consulenza IT, intende sfruttare la tecnologia blockchain al fine di migliorare l'affidabilità delle aziende e delle imprese a vantaggio di investitori, creditori, fornitori, parti interessate e consumatori. La Blockchain, per le sue caratteristiche, rappresenta infatti uno strumento moderno, efficiente e affidabile per la consulenza aziendale, la governance e la finanza. Il presente lavoro elabora l'applicazione della tecnologia Blockchain al settore agro-alimentare, con particolare attenzione al settore vitivinicolo. Si intende realizzare un prototipo di un'applicazione decentralizzata basata su Ethereum e pensata per la cantina del novarese "Podere ai Valloni". L'intenzione dell'azienda è ricorrere alla tecnologia blockchain per garantire la trasparenza, la qualità e la territorialità del proprio vino, in maniera tale da rispondere alle esigenze di una domanda sempre più attenta alle tematiche ambientali e per migliorare la comunicazione con gli stakeholder.

L'obiettivo è dimostrare la fattibilità del progetto di Sblockchain Project e ottenere un feedback iniziale dal cliente.

Il lavoro di tesi è così articolato:

- il secondo Capitolo tratta l'analisi sullo stato dell'arte della Blockchain e la sua applicazione al settore-agroalimentare, in particolare, a quello vitivinicolo. Dopo aver brevemente illustrato le origini della tecnologia blockchain e sua successiva affermazione, ci si sofferma sulla rilevanza che tale tecnologia ha oggi nella ricerca scientifica. Confermata la crescente domanda di trasparenza nel settore agro-alimentare da parte dei consumatori e analizzate le frodi alimentari più diffuse, si passa a un'analisi di vantaggi e svantaggi che ricorrere

a tale tecnologia può comportare, quindi ci si sofferma sul settore vitivinicolo e sulle applicazioni già esistenti. Inoltre, per dimostrare che ricorrere all'applicazione della tecnologia blockchain può essere economicamente vantaggioso per un'impressa vitivinicola e, in particolare, per una PMI, viene riportato il caso della Cantina Placido Volpone, prima cantina al mondo a certificare la propria filiera tramite blockchain;

- il terzo Capitolo contiene una descrizione del caso di studio e un'analisi dei requisiti volta ad identificare le richieste e le necessità dell'azienda che il sistema deve soddisfare;

- il quarto Capitolo ha come tema principale le tecnologie utilizzate e le motivazioni che hanno portato alla scelta e all'utilizzo di determinate tecnologie. Questo permette di fornire le conoscenze necessarie alla comprensione del funzionamento del prototipo realizzato;

- il quinto Capitolo parte dall'architettura del sistema per arrivare a una descrizione dettagliata del prototipo realizzato, includendo sia il back-end che il front-end e fornendo delle schermate esplicative del sistema;

- l'ultimo Capitolo, infine, il sesto, è dedicato a una valutazione dei risultati ottenuti e di eventuali opportunità per sviluppi futuri.

Capitolo 2

Stato dell'arte

Il lavoro di tesi ha come tema principale la tecnologia *blockchain* e la sua applicazione nel settore agro-alimentare, con particolare attenzione a quello vitivinicolo. L'obiettivo di questo Capitolo è fornire una fotografia della situazione attuale della Blockchain e della sua applicazione.

2.1 Blockchain

2.1.1 Nascita e sviluppo della tecnologia

Nel presente paragrafo vengono elencate le tappe fondamentali dell'invenzione e dell'evoluzione della Blockchain.

1991: *Stuart Haber e Scott Stornetta co-inventano la prima blockchain.*

Alla Bell Communications Research (Bellcore), Scott e Stuart realizzano una serie pionieristica di documenti e brevetti che hanno gettato le basi per Bitcoin e altre valute digitali. Infatti, delle otto citazioni nel white paper originale di Bitcoin [1], tre fanno riferimento al loro lavoro. I due scienziati cercano di

dare una risposta a un importante problema: come mantenere le informazioni digitali al sicuro da alterazioni?

> *In this paper we have shown that the growing use of text, audio, and video documents in digital form and the ease with which such documents can be modified creates a new problem: how can we certify when a document was created or last modified?* [2]

Pubblicano «How to time-stamp a digital document» in cui evidenziano i limiti del ricorrere a un ente «certificatore» (TTS – Time Stamping Service) e introducono l'uso di crittografia, firma digitale e catene di blocchi per la marcatura temporale di documenti digitali. Co-inventano così la tecnica blockchain per garantire l'integrità dei record digitali. Inoltre hanno successivamente co-fondato Surety Technologies, uno spin-off di Bellcore che offriva servizi di timestamp digitale. È stata la prima implementazione commerciale di una Blockchain e ha avuto inizio nel 1994 [3].

1992: *Nasce l'idea della Proof of Work.*

Nel 1992 Cynthia Dwork e Moni Naor pubblicano "Pricing via Processing or Combatting Junk Mail"[4] con l'obiettivo primario di presentare una tecnica computazionale per combattere la posta indesiderata e, più in generale, per controllare l'accesso a una risorsa condivisa. L'idea principale è quella di richiedere a un utente di calcolare una funzione moderatamente difficile per ottenere l'accesso alla risorsa, impedendo un uso frivolo.

Dwork e Naor introducono così uno dei meccanismi di consenso più noti alla base di varie blockchain, il *"Proof of Work"* (PoW).

2008: *Satoshi Nakamoto definisce Bitcoin.*

Satoshi Nakamoto (pseudonimo usato dal creatore o dai creatori di Bitcoin) pubblica il whitepaper "Bitcoin: A Peer-to-Peer Electronic Cash System"[1] in cui vengono definiti gli aspetti centrali (hashing, marcatura temporale, proof of work) di una delle più note blockchain: Bitcoin.

Con il documento è stato inventato un sistema di pagamento totalmente elettronico basato su una rete peer-to-peer, indipendente da autorità centrali come le istituzioni finanziarie. L'anno successivo alla pubblicazione dell'articolo, nasce la prima blockchain chiamata, come suggerisce il titolo del white paper, *Bitcoin*. Esso prevede lo scambio di valuta virtuale senza ricorrere a terzi e tra pari. Infatti, in una tipica transazione, una terza parte deve essere coinvolta nel trasferimento di denaro ma le terze parti non sono sempre affidabili: c'è la possibilità che vengano compromesse, e, inoltre, possono esserci dei limiti al trasferimento e importi addizionali. Bitcoin, invece,è immune da contraffazioni ed è preservato da algoritmi complessi [5].

2013: *Nascita di Ethereum.*

Ethereum è stato introdotta nel 2013 con un white paper[1].

L'autore è Vitalik Buterin, sviluppatore di origini russe, cresciuto in Canada, programmatore di bitcoin e co-fondatore della rivista "Bitcoin Magazine" e proponeva un'evoluzione del protocollo bitcoin. Altri appassionati di criptovalute, attratti dall'idea, si sono uniti al progetto. Il team di sviluppo di Bitcoin, però, non supporta il progetto: decidono così di creare una nuova piattaforma, Ethereum.

[1] "Ethereum: A Next-Generation Smart Contract and Decentralized Application Platform", Vitalik Buterin, 2014

Ethereum è una piattaforma decentralizzata del Web 3.0 per la creazione e pubblicazione peer-to-peer di contratti intelligenti (smart contract). La criptovaluta a essa legata è l'Ether.

La Ethereum Virtual Machine prevede i cosiddetti contratti intelligenti, o programmi che possono essere distribuiti sulla Blockchain, consentendo agli sviluppatori di creare dApp (applicazioni decentralizzate) che vengono collocate e distribuite sulla catena stessa [3].

2014: Nell'agosto 2014, la dimensione del file blockchain di Bitcoin, contenente le registrazioni di tutte le transazioni avvenute sulla rete, ha raggiunto i 20 GB.

2015: Nel gennaio 2015, la dimensione è cresciuta fino a quasi 30 GB.

2016-oggi: Nel 2016 viene usato per la prima volta il termine *"Blockchain"*, nato dall'unione delle parole *blocco* e *catena*, usate separatamente nel documento originale di Satoshi Nakamoto.

Dal gennaio 2016 al gennaio 2017, la blockchain di Bitcoin è passata da 50 GB a 100 GB di dimensione. All'inizio del 2020, le dimensioni del libro mastro avevano superato i 200 GB per arrivare agli attuali 450 GB[2](Fig. 2.1) [6].

[2]https://www.blockchain.com/explorer/charts/blocks-size

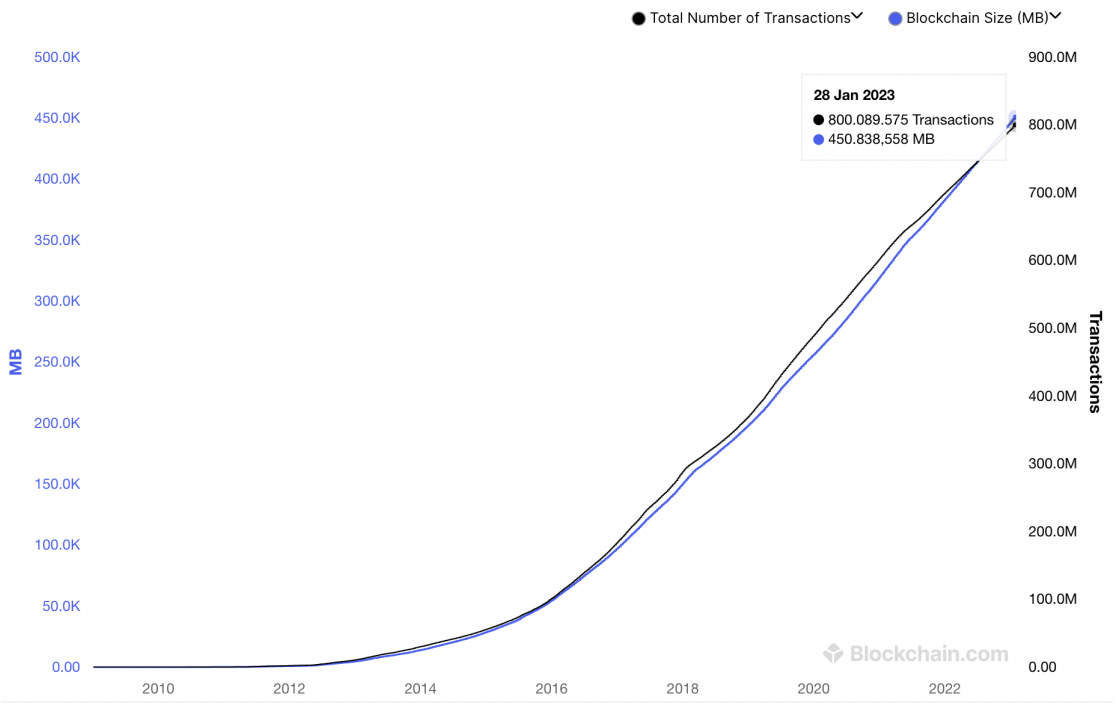

Figura 2.1: Dimensione della Blockchain e numero di transazioni in data 28 Gennaio 2023.

2.1.2 Definizione

La blockchain è un registro (ledger) digitale distribuito in grado di memorizzare transazioni in modo sicuro, verificabile e permanente. Questa si basa su una rete Peer-to-Peer (P2P) costituita da tutti quei nodi che leggono o scrivono in modo cooperativo le transazioni nella blockchain. Il termine deriva dalla sua struttura tecnica: una catena di blocchi dove ogni blocco è collegato al blocco precedente con un hash crittografico. Un blocco è una struttura dati che consente di memorizzare un elenco di transazioni, le quali sono create e scambiate da peer della rete blockchain e modificano lo stato della stessa.

Inizialmente la blockchain era esclusivamente la tecnologia di gestione decentralizzata di Bitcoin, progettata per l'emissione e il trasferimento di denaro per gli

utenti della valuta Bitcoin. Questa tecnica poteva supportare il registro pubblico di tutte le transazioni Bitcoin, senza alcun controllo da parte di una Third Party Organization[3]. Il vantaggio della Blockchain è che il libro mastro pubblico non può essere modificato o cancellato dopo che i dati sono stati approvati da tutti i nodi [7]. Date le sue caratteristiche di integrità e sicurezza dei dati, negli anni la sua applicazione è stata estesa sempre a più settori: risulta applicabile agli ambienti in cui è necessario effettuare delle transazioni, non solo quelle relative alle criptovalute. Attualmente il campo di applicazione della blockchain risulta essere così ampio da includere: Supply Chain Management, E-Voting, Smart-Grid, Healthcare, Banking, Smart Cities, Vehicular, andando ben oltre le criptovalute originali come Bitcoin.

2.1.3 Fondamenti

Prima di procedere oltre, vengono elencati e definiti alcuni termini utilizzati nel corso del presente documento in modo da renderlo di più facile lettura e comprensione.

Ledger: il registro pubblico nel quale vengono «annotate» in maniera trasparente e immutabile le transazioni. É composto dai blocchi concatenati tra loro tramite una funzione di hash.

Funzione di hash: funzione che mappa dei dati in una stringa univoca di dimensione fissa.

Firma digitale: la tipica firma digitale prevede due fasi, una fase di firma e una fase di verifica. Ogni utente possiede una coppia di chiavi: una chiave privata, che deve essere mantenuta segreta e che viene utilizzata per firmare le transazioni, e una chiave pubblica. La blockchain utilizza un meccanismo

[3]qualsiasi cliente, fornitore, partner, agente, appaltatore principale, subappaltatore o altra organizzazione attuale o potenziale

di crittografia asimmetrica[4] per convalidare l'autenticazione delle transazioni, tipicamente questo tipo di crittografia viene utilizzata in contesti non affidabili.

Nodi: partecipanti alla blockchain connessi in rete (sono fisicamente i server di ciascun partecipante).

Blocchi: insieme di transazioni raggruppate per la verifica, l'approvazione e l'archiviazione da parte dei partecipanti alla blockchain. Un blocco è composto da due parti principali: l'header e il body. Le transazioni sono racchiuse nel body del blocco e nell'header sono presenti i campi di gestione del blocco stesso come mostrato nella Figura sottostante (Fig. 2.2).

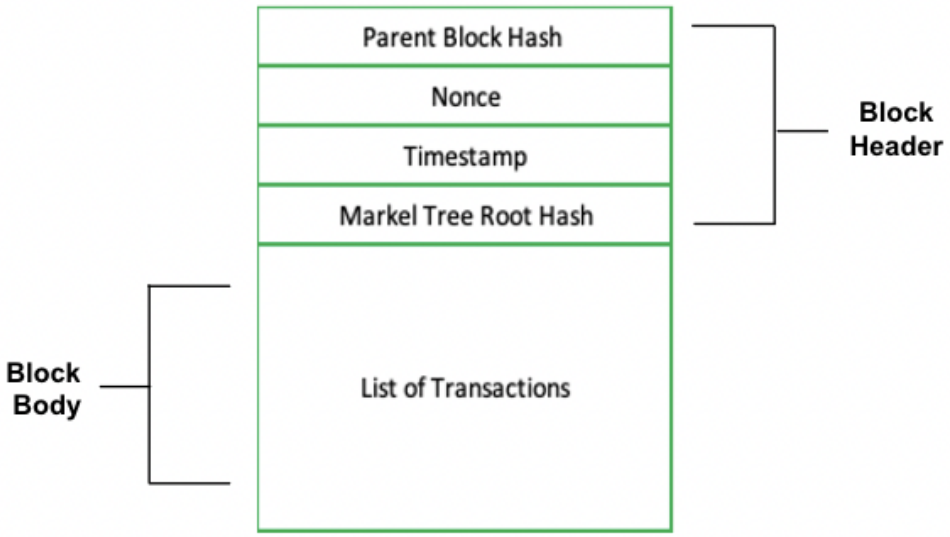

Figura 2.2: Struttura generica di un blocco.

Il *block header* si compone di:

[4]La crittografia asimmetrica si riferisce a un tipo di crittografia in cui la chiave utilizzata per crittografare i dati è diversa da quella utilizzata per decifrare i dati.

1. *Parent block hash*: un valore hash a 256 bit che punta al blocco precedente;
2. *Nonce*: è un numero che viene generato e utilizzato una sola volta. É un campo di 4 byte, che di solito inizia con 0 e aumenta per ogni calcolo dell'hash;
3. *Timestamp*: marcatore temporale che viene assegnato ad ogni transazione;
4. *Markel tree root hash*: il valore hash di tutte le transazioni del blocco.

Il *block body*, invece, è costituito della *lista delle transazioni*, ossia i dati da memorizzare nella blockchain che devono essere sottoposti a processo di verifica, approvazione (consenso) e archiviazione. Il numero massimo di transazioni che un blocco può contenere dipende dalla sua dimensione e dalla dimensione di ciascuna transazione. Al verificarsi di una nuova transazione, viene aggiunto un nuovo blocco con i dati corrispondenti. Si crea così una catena, da cui il nome *Blockchain* (Figura 2.3).

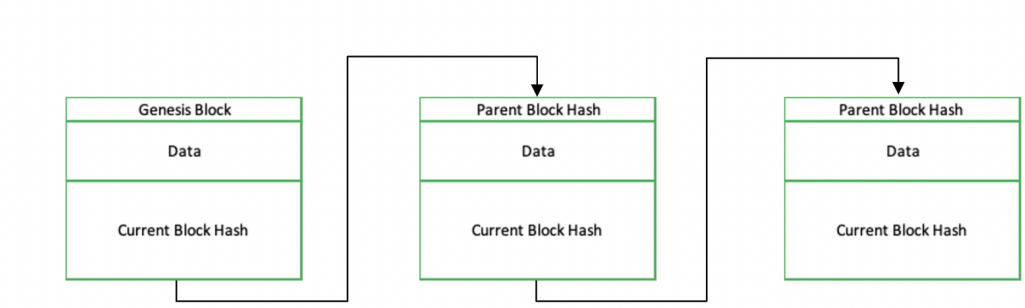

Figura 2.3: Catena di blocchi.

Il primo blocco prende il nome di *Genesis Block* e non non avrà il valore hash di nessun blocco precedente. A causa della sua natura immutabile, i dati salvati nella Blockchain non possono essere modificati. Infatti, quando i dati di un blocco vengono modificati, il suo valore hash cambia quindi non sarà più corrispondente

al parent block hash del blocco successivo, il che interrompe la catena e la invalida. Per evitare ciò, vengono utilizzati i modelli di consenso.

2.1.4 Modelli di consenso

Un protocollo di consenso è un insieme di politiche concordate e attuate da tutti i nodi, ossia un insieme di regole che disciplinano quali e come nuove transazioni possono essere aggiunte alla blockchain [8]. Il processo di consenso [9] permette la lettura e l'aggiornamento dello stato condiviso che assicura l'ordine delle transazioni e garantisce l'integrità dei contenuti in modo decentralizzato. Diverse blockchain impiegano diversi modelli di consenso, tra cui Proof-of-Work e Proof-of-Stake (PoS). In generale, i protocolli di consenso vengono selezionati sulla base di tre proprietà essenziali: sicurezza, vitalità e tolleranza ai guasti.

Proof of Work

Per aggiungere blocchi alla catena, una *proof-of-work* deve essere comunicata. Bitcoin utilizza il concetto di PoW come meccanismo di consenso, che scala oltre 1000 nodi. La PoW[9] richiede che l'iniziatore risolva un'operazione matematica o crittografica mediante il metodo forza bruta[5], e produca un valore,detto anche *winning value*, che sia inferiore a un valore definito dalla rete. Può accadere che più di un nodo produca un valore vincente nello stesso momento: questa situazione crea una biforcazione che viene risolta dalla rete analizzando il valore massimo del proof-of-work, ovvero il lavoro massimo svolto da un nodo. La richiesta di aggiornamento da parte del nodo con la prova di lavoro minima viene scartata. In questo modo viene garantita la coerenza dello stato tra tutti i nodi. La PoW si

[5]Nella sicurezza informatica, indica un algoritmo di risoluzione di un dato problema che consiste nel verificare tutte le soluzioni teoricamente possibili fino a che si trova quella effettivamente corretta.

adatta meglio alle reti che richiedono scalabilità, infatti, la maggior parte delle *blockchain permissionless* utilizzano PoW in quanto hanno l'autenticità del nodo partecipante, di conseguenza le dimensioni della rete diventano molto grandi.

La Proof-of-Work presenta però anche significativi svantaggi in quanto:

- richiede che ogni nodo investa ingenti somme per l'acquisto delle attrezzature utilizzate nel processo di mining;
- supporta una velocità di transazione molto bassa, pari a solo 7 transazioni al secondo;
- richiede un notevole dispendio di energia e un'elevata latenza.

Quindi, se Proof-of-Work è stato per anni il meccanismo di consenso prevalente, a causa del grande consumo energetico, le blockchain più recenti implementano algoritmi più ecologici ma piuttosto centralizzati, come la Proof-of-Stake (PoS) e la Delegated Proof of Stake (DPoS). Più recentemente, altre blockchain hanno implementato i propri protocolli ibridi o ad hoc [10].

Proof of Stake

Proof of Stake sostituisce il meccanismo di mining del modello PoW, che utilizza elevate quantità di energia per i calcoli. Invece di acquistare, ad esempio, attrezzature per generare valori vincenti, PoS suggerisce di acquistare criptovalute e di usarle per comprare le possibilità di creazione di blocchi nella blockchain; elimina i minatori e li sostituisce con i *validatori*. Significativo è il recente passaggio di Ethereum dal meccanismo POW al POS in quanto utilizzare il meccanismo Proof-of-Work, oltre agli inconvenienti sopra citati, generava anche rifiuti elettronici. I server informatici specializzati, utilizzati per il mining di criptovalute, spesso diventavano obsoleti in 1,5 anni per poi finire nelle discariche. Pianificato da anni, ma avvenuto solo nel settembre del 2022, il "The Merge" [11] ha formalizzato il passaggio dal

meccaniscmo di consenso Proof-of-Work al Proof-of-Stake. Ethereum ha trasferito l'intera rete a un diverso meccanismo di consenso che utilizza il 99% in meno di energia, consentendo alla rete di scalare e di aumentare il numero di transazioni al secondo.

Figura 2.4: Consumo Energetico Ethereum per anno.

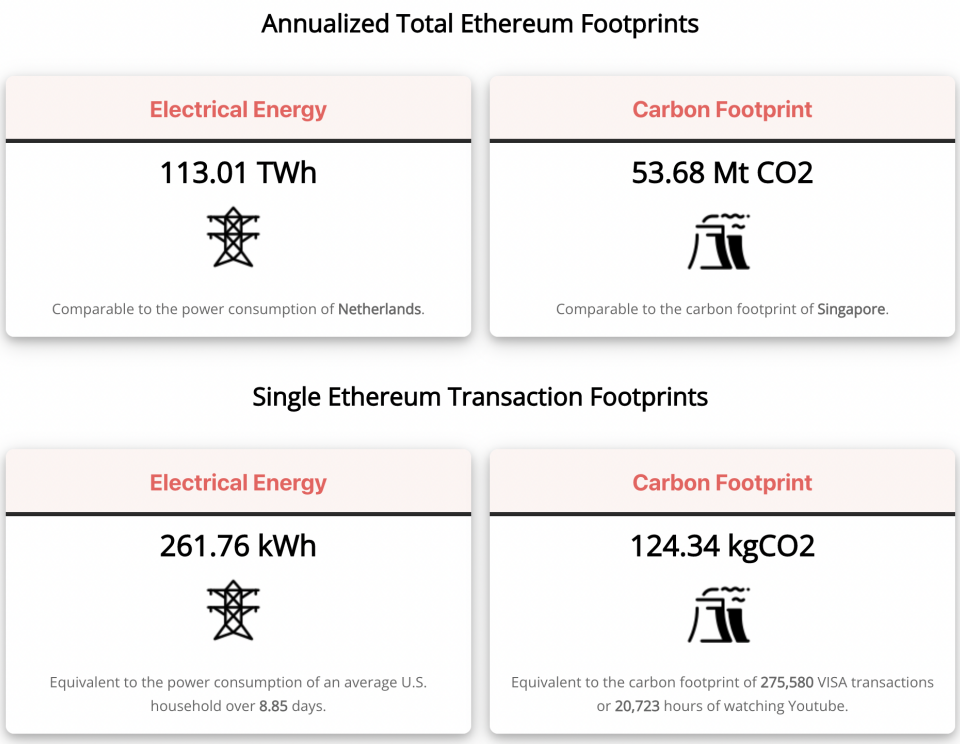

Figura 2.5: Consumo Energetico Ethereum totale e per transazione.

Come esplicitato dalle Figure 2.4 e 2.5, tratte da *Digiconomist*[6] e risalenti a febbraio del 2022 (prima del passaggio), Ethereum, prima del passaggio, consumava 113 terawattora all'anno, una quantità di energia pari a quella dei Paesi Bassi. Una singola transazione, invece, arrivava a consumare tanta energia quanta ne consuma una famiglia media statunitense in più di una settimana.

Ethereum, oggi, utilizza la proof-of-stake, in cui i validatori puntano esplicitamente un capitale sotto forma di ETH in uno smart contract su Ethereum. L'ETH puntato funge da garanzia che può essere distrutta se il validatore si comporta in modo disonesto o pigro. Il validatore è quindi responsabile di controllare che i nuovi blocchi propagati sulla rete siano validi e, occasionalmente, di creare e propagare lui stesso nuovi blocchi. Inoltre, il gruppo di entità autorizzate a scrivere nuove transazioni nello Shared Ledger, aggiungendole alla blockchain, può variare da pochi utenti selezionati e autenticati fino a qualsiasi utente anonimo. Questi diversi privilegi di scrittura dipendono dalle regole del protocollo di consenso scelto, che sono decisive per determinare se iln ledger condiviso risultante sarà pubblico o privato [8].

2.1.5 Tipologie di Blockchain

Le industrie hanno iniziato a cercare di rinnovare i loro modelli di business per trarre vantaggio da questa nuova tecnologia. La Blockchain può essere utilizzata in tre tipi di ambienti di implementazione[12]:

Permissionless o Public Blockchain

Una Blockchian Permissionless è un ambiente open-source a cui chiunque può accedere o partecipare. L'accesso avviene senza permessi: gli attori della rete

[6]https://digiconomist.net/

sono anonimi, chiunque può effettuare transazioni, visualizzarne l'intera cronologia e partecipare al meccanismo di consenso. Il vantaggio delle reti pubbliche è che nessun individuo o entità è in grado di controllare le informazioni sul libro mastro e, pertanto, il sistema è neutrale. Gli svantaggi,invece, riguardano la privacy e la scalabilità: essendo i partecipanti sconosciuti, possono esserci attori malintenzionati nella rete. Pertanto, ci possono essere applicazioni in cui alcuni tipi di informazioni sono troppo sensibili per essere condivisi in un libro mastro completamente aperto, come nel caso delle istituzioni finanziarie. Allo stesso modo, la blockchain presenta limiti di scalabilità in riferimento alla dimensione dei dati, alla velocità di elaborazione delle transazioni e soffre di latenza nella trasmissione dei dati. Le due piattaforme pubbliche più conosciute sono la blockchain di Bitcoin ed Ethereum.

Permissioned Blockchain

Una blockchain permissioned si differenzia da una permissionless per il controllo degli accessi. Questo ambiente prevede reti proprietarie (ovvero private o chiuse) che definiscono e decidono i partecipanti e i loro ruoli nella rete ed è sviluppato principalmente dalle industrie per il loro uso commerciale privato. Essa limita gli utenti in termini di accesso al meccanismo di consenso e quindi consente solo ai partecipanti previsti di entrare nella rete diversamente dalle blockchain permissionless, alle quali può aderire qualsiasi utente. Inoltre, richiedendo che l'identità degli attori sia nota nel libro mastro distribuito privato o autorizzato, vi è un ulteriore livello di sicurezza che limita gli attori malintenzionati, in quanto possono essere penalizzati ed espulsi dalla rete. Invece di partecipanti anonimi, i libri mastri distribuiti autorizzati utilizzano persone giuridiche già autenticate per convalidare le transazioni. Questo li rende più interessanti per i mercati globali dei capitali, i beni tangibili e le catene di approvvigionamento [13]. Le DLT private

utilizzano due gruppi principali di algoritmi di consenso: quelli basati sulla lotteria[7] e quelli basati sul voto[8]. Gli algoritmi *lottery-based* sono vantaggiosi in termini di scalabilità, ma comportano tempi più lunghi per il raggiungimento del risultato finale, mentre gli algoritmi *voting-based* sono vantaggiosi in termini di velocità e di risultato finale, ma non sono scalabili.

Hybrid o Consortium Blockchain

In realtà esiste anche una terza categoria, nota come catena di blocchi ibrida o consortile. Deriva da due dei tipi di Blockchain di base sopra menzionati. Ad esempio, una blockchain ibrida può essere una permissionless che affida la validazione delle transazioni a un numero ristretto di nodi. In generale, il funzionamento del sistema è flessibile: la visibilità della catena può essere limitata ai validatori, visibile a persone autorizzate o a tutti. Un consorzio blockchain è la soluzione adatta in un contesto in cui diverse organizzazioni operano nello stesso settore e richiedono un piano comune su cui eseguire transazioni o trasmettere informazioni. Partecipare a un consorzio di questo tipo potrebbe essere vantaggioso per un'organizzazione, in quanto consentirebbe di condividere informazioni sul settore con altri operatori.

2.1.6 Diffusione della tecnologia

La tecnologia Blockchain [14] è diventata sempre più popolare negli ultimi decenni, come testimoniato dall'elevato numero di articoli scientifici sull'argomento. È stato utilizzato il database di Web of Science[9] per recuperare record bibliografici attraverso la parola chiave "Blockchain" nel campo *topic*. É stato inoltre applicato

[7]dipende dal modello di elezione a sorte per selezionare un leader nella rete (ad esempio, pow e pos)

[8]meccanismi di voto per finalizzare uno stato comune

[9]servizio di indicizzazione di citazioni scientifiche

un filtro sulla data: sono stati presi in considerazione tutti i documenti dal 1970 al 2023.

Svolgendo la ricerca per il topic blockchain fino a febbraio del 2023, si ottengono un totale di 27,610 pubblicazioni. La suddivisione delle pubblicazioni per tipologia di documento, come riportato in Appendice A.1, evidenzia come, delle diciannove tipologie di documenti individuate, il maggior numero di pubblicazioni riguardi la tipologia di documento *article* (il 53.495% del totale), seguito dal *preceeding paper* (il 39.786%) e dal *review article* (il 4.694%). Le restanti sedici costituiscono solo il 2.025%del totale. Quindi, le categorie di documenti dominanti riguardanti la blockchain risultano essere gli articoli e i preeciding paper con il 93.281% del totale.

I risultati ottenuti analizzando la diffusione per area geografica sono riportati di seguito in forma tabulare e, in maniera più estesa, in Appendice (Figura A.2).

Area Geografica	**Numero Pubblicazioni**	**% su 27610**
Cina	8,521	30.862%
USA	4,587	16.614%
India	2,683	9.717%
Regno Unito	1,904	6.896%
Australia	1,558	5.643%
Sud Corea	1,429	5.176%
Canada	1,306	4.730%
Germania	1,205	4.364%
Italia	1,202	4.353%
Arabia Saudita	952	3.448%

Tabella 2.1: Numero di documenti con "Blockchain" come topic su WebOfScience: suddivisione per area geografica.

Dai risultati ottenuti si evince come molta attenzione alla ricerca e alla tecnologia venga data principalmente in Cina e negli Stati Uniti D'America, che, da soli, ricoprono il 47.476% delle pubblicazioni totali. In tale classifica, l'Italia si colloca tra le prime 10 risulta con 1202 pubblicazioni sul tema.

Per fornire un quadro più completo, si è voluto anche analizzare l'andamento del numero di pubblicazioni rispetto agli anni considerati. Come specificato in precedenza, il database è stato impostato per considerare le pubblicazioni dal 1970 ad oggi. Secondo WebofScience, prima del 2015 c'erano solo 12 pubblicazioni riguardati la blockchain ma il numero aumenta notevolmente dal 2018 in poi. Solo tra il 2017 e il 2018 il numero di pubblicazioni è più che triplicato (da 677 a 2,562).

Anno	**Numero Pubblicazioni**	**% su 27610**
2022	7,189	26.038%
2021	6,505	23.560%
2020	5,605	20.301%
2019	4,432	16.052%
2018	2,562	9.279%
2017	677	2.452%
2016	146	0.529%
2015	25	0.091%
2014	10	0.036%
2013	2	0.007%

Tabella 2.2: Numero di documenti con "Blockchain" come topic su WebOfScience: suddivisione per anno.

Nella Tabella si evidenzia un chiaro andamento crescente (nell'analisi è stato

escluso l'anno 2023 in quanto non ancora terminato) che conferma l'interesse sempre maggiore per la tecnologia. In Figura A.3 i dati vengono riportarti sotto-forma di istogramma. L'interesse crescente per la Blockchain indica chiaramente che sta diventando sempre più rilevante per il futuro. Infatti, è in grado di fornire una soluzione decentralizzata e sicura per la memorizzazione e la condivisione di dati, che rappresenta una sfida per molte delle attuali tecnologie centralizzate.

2.1.7 La Blockchian nella Supply Chain

In generale, la blockchain rappresenta una tecnologia promettente che offre molte opportunità di innovazione e sviluppo in numerosi settori.

La blockchain presenta alcune caratteristiche uniche, quali [15]:

- **decentralizzazione**: è una tecnologia decentralizzata peer-to-peer, chiunque può accedere facilmente ai dati;
- **immutabilità e integrità dei dati**: ogni volta che una nuova transazione viene registrata nel database computerizzato della blockchain, non può essere alterata, i dati non possono essere modificati senza il consenso della rete;
- **sicurezza**: ad ogni transazione viene assegnato un codice hash crittografico univoco e temporizzato, che può essere un valore di firma alfanumerico a 64 o 128 cifre;
- **affidabilità**: la decentralizzazione della rete permette a ogni partecipante di controllare l'intera catena, rendendo il sistema più sicuro e riducendo il rischio di intromissione da parte di malintenzionati;
- **trasparenza**: tutte le transazioni coinvolte sono trasparenti in quanto tutti possono vederne i dettagli e ogni nodo contiene il libro mastro completo. Il libro mastro digitale condiviso contiene tutte le informazioni sull'origine, la destinazione, la data e l'ora delle transazioni in blocco.

Grazie a queste caratteristiche, la blockchain ha numerosi campi di applicazione. Per questo lavoro di tesi viene presa in considerazione la sua applicazione nella supply chain.

Secondo lo studio condotto da Maher A.N. Agi e Ashish Kumar Jha [16], il vantaggio relativo della tecnologia e la pressione esterna sono i fattori più importanti che influenzano l'adozione di blockchain nella catena di fornitura. Anche il potenziale della blockchain di ridurre i costi delle transazioni, l'interesse dei consumatori per i dati di tracciabilità e la creazione di un quadro normativo per l'utilizzo della blockchain hanno un ruolo causale significativo nell'adozione.

Nel contesto della supply chain, la tecnologia blockchain vine impiegata per:

- *tracciabilità dei prodotti*, le informazioni sul prodotto possono essere rintracciate attraverso la catena di fornitura;
- *logistica*, viene utilizzata per acquisire informazioni sul movimento fisico delle merci in relazione all'automazione, alla gestione della catena di fornitura e dell'inventario, all'uso dell'IoT e della catena del freddo e alla registrazione della storia ambientale;
- *transazioni finanziare*, viene utilizzata per i pagamenti, in particolare con l'uso di contratti intelligenti e token;
- *operazioni di vendita al dettaglio*, anche in questo caso la blockchain viene utilizzata per i pagamenti, con l'uso di contratti intelligenti e token;
- *economia circolare*, utilizza la blockchain per l'uso secondario dei beni e la rivendita, con lo scopo di riutilizzare e riciclare.

La ricerca *"An Analysis of Blockchain Adoption in Supply Chains Between 2010 and 2020"* [17], analizzando 271 progetti e sfruttando parametri quali data di inizio, tipo di blockchain, settori applicati e tipologia di organizzazione, mappa l'evoluzione

dell'applicazione della tecnologia blockchain alla catena di approvvigionamento fino a giugno 2020. Dallo studio si evince che la maggior parte dei progetti che riguardano l'applicazione della blockchain alla supply chain, sono sulla tracciabilità del prodotto. Un progetto può appartenere a più aree applicative.

Area	**Numero Progetti**	**% su 271**
Tracciabilità del prodotto	180	66.41%
Logistica	120	44.28%
Transazioni finanziarie	66	24.35%
Economia circolare	21	7.75%
Vendita al dettaglio	18	6.64%

Tabella 2.3: Numero di progetti per una particolare area della supply chain.

In conclusione, l'uso più evidente della blockchain è quello di coordinare e tracciare le informazioni: la sicurezza alimentare è di fondamentale importanza, così come la capacità di tracciare e rintracciare i prodotti agricoli e alimentari.

2.2 Il settore agroalimentare

Secondo la legislazione Europea, per "tracciabilità" nel settore agroalimentare si intende la possibilità di rintracciare qualsiasi alimento, mangime, animale destinati alla produzione alimentare o di un ingrediente attraverso tutte le fasi di produzione, trasformazione e distribuzione [18].

2.2.1 La richiesta di trasparenza

Il rapporto del Food Marketing Institute e di Label Insight [19] del 2018, sostiene la necessità di trasparenza nel futuro del settore agro-alimentare. Offre, inoltre, delle raccomandazioni alle aziende del settore. Viene evidenziata l'importanza della trasparenza per aumentare la fedeltà nei clienti e per promuovere la fiducia dei consumatori: i clienti richiedono sempre più informazioni e non si accontenteranno della lista degli ingredienti. Chiedono ai produttori anche informazioni sugli effetti collaterali, le allergie, gli standard di qualità e la conservazione. Per sopravvivere sul mercato, è necessario che le aziende rispondano alle richieste dei consumatori e, se necessario, che apportino rapidi aggiustamenti e cambiamenti strategici. Infatti, il 75% dei consumatori dichiara che passerà a un'azienda che offre dettagli sul prodotto che vanno oltre quanto stampato sull'etichetta vera e propria. Solo due anni prima, nel 2016, il 39% dei consumatori a cui è stata posta la stessa domanda ha dichiarato che cambierebbe marca; questo testimonia la crescente richiesta di trasparenza da parte del consumatore. Secondo lo studio, il 61% delle famiglie americane con membri che seguono regimi dietetici o di fitness è disposto a pagare di più per prodotti con informazioni complete sul prodotto, l'89% di queste famiglie ha dichiarato che passerà a un prodotto diverso se non è soddisfatto delle informazioni disponibili.

A tal proposito, nel review article di Vinay Singh e Sanjeev Kumar Sharma del 2021 [20], vengono classificate le frodi alimentari in quattro categorie:

1. *etichettatura errata*: quando i prodotti vengono venduti, possono essere incluse deliberatamente false informazioni sulla confezione per fuorviare i consumatori;

2. *documentazione falsa*: può accadere che documenti e certificazioni siano assenti, oppure che i relativi dati vengano modificati;

3. *violazione dei diritti di proprietà intellettuale*: con cui si intende qualsiasi l'imitazione di prodotti autentici (anche nota come "contraffazione");

4. *ingredienti impropri*: consiste nell'aggiunta, nella rimozione o sostituzione di ingredienti per ottenere un vantaggio economico.

2.2.2 L'introduzione della blockchain

Nonostante i processi produttivi siano quasi sempre digitalizzati con tecnologie come il cloud computing, l'intelligenza artificiale e l'Internet of Things(IoT), ricorrere alla blockchain permette di aumentare l'efficienza e la trasparenza delle filiere. Le sue caratteristiche peculiari quali decentralizzazione, trasparenza, sicurezza e immutabilità, ne consentono la piena applicazione nel settore agroalimentare, con un duplice beneficio: la Blockchain permette alle imprese di certificare la qualità e la sostenibilità dei processi produttivi, aumentando la sicurezza percepita da parte dei consumatori e, ai consumatori, di esplorare l'origine del prodotto, le tecniche colturali e le trasformazioni subite dal prodotto stesso.

Le sfide del settore agroalimentare, su piccola e larga scala, non riguardano solo la mancanza di trasparenza ma anche l'alta percentuale di lavoro manuale e cartaceo, la mancanza di interoperabilità e le informazioni limitate sulla tracciabilità del prodotto.

L'applicazione della blockchain fornisce un database digitale che registra, traccia e monitora beni fisici e digitali, consente transazioni di qualità superiore e una maggiore tracciabilità. La tecnologia può integrare e gestire in tempo reale ogni processo e transazione lungo la filiera agricola. Ogni transazione elaborata sul libro mastro distribuito può contenere i dettagli della transazione e gli attributi specifici del prodotto che possono essere aggiunti dagli attori della filiera. Gli attori possono identificare ed esaminare il movimento del prodotto lungo ogni fase della filiera, dagli input agricoli e zootecnici (fertilizzanti, foraggi, ecc.) utilizzati nell'azienda

agricola fino al rivenditore. La blockchain memorizza record immutabili che sono trasparenti e, in teoria, accessibili a qualsiasi utente dotato di software. Questa tecnologia ha il potenziale per creare grandi guadagni di efficienza per ogni attore della catena di approvvigionamento. Offre, inoltre, anche una reale opportunità di partecipazione al mercato più inclusiva per i piccoli proprietari e le PMI perchè permette di ridurre i costi, di aumentare la competitività e migliorare il rapporto con il consumatore [21].

I maggiori benefici della blockchain riguardano le realtà caratterizzate da una maggiore frammentazione e complessità della filiera perché fornisce un maggiore controllo lungo l'intera catena del valore. Ci sono numerosi altri vantaggi, oltre quelli già evidenziati in precedenza, apportati dal ricorso alla tecnologia:

- in caso di focolaio di una malattia animale o vegetale, prodotti agroalimentari contaminati o frode alimentare, la blockchain consente alle imprese e alle autorità di regolamentazione di rintracciare e individuare i prodotti contaminati o fraudolenti in modo più rapido e meno dispendioso;

- può contribuire alla mitigazione del cambiamento climatico, porterebbero un approccio più trasparente ed efficiente alla contabilizzazione e alla compensazione del carbonio;

- l'applicazione delle DLT nelle catene di approvvigionamento agricolo, nei registri catastali e nei servizi finanziari può aiutare il settore pubblico a raggiungere i propri obiettivi di politica pubblica per la sicurezza alimentare e lo sviluppo rurale ed essere uno stimolo per raggiungere gli SDG.

La blockchain presenta però anche delle limitazioni, tra queste:

- l'elevato consumo energetico che questi sistemi pagano per fornire una potenza di calcolo hardware estesa e le esigenze dell'infrastruttura di rete;

- il costo della tecnologia che potrebbe comportare un aumento del prezzo dei prodotti dotati di tracciabilità digitale;
- la sua complessità rappresenta una potenziale sfida per una comprensione diffusa della tecnologia.

L'aumento del prezzo del prodotto finale non sembra però turbare i consumatori: secondo lo studio *"Are the Innovative Electronic Labels for Extra Virgin Olive Oil Sustainable, Traceable, and Accepted by Consumers?"* [22] condotto in Italia, i consumatori sono disposti a pagare, per l'integrazione delle tecnologie di tracciabilità, un prezzo aggiuntivo pari al 17,8% rispetto al prezzo base.

Oltre a valutare i costi, potrebbe essere utile porsi ulteriori domande, come se ricorrere alla tracciabilità tramite blockchain possa compromettere l'immagine che i consumatori hanno del prodotto. Ad esempio, alcuni prodotti italiani, tra cui alcuni alimenti a denominazione di origine protetta, sono realizzati con materie prime provenienti da Paesi geograficamente lontani che potrebbero compromette l'"italianità" del prodotto [14]. É il caso della Breasaola della Valtellina, un alimento italiano che spesso viene realizzato con carni provenienti da Paesi diversi. Questo non significa che l'alimento sia di scarsa qualità ma che bisogna chiedersi se c'è il rischio di rendere il prodotto "meno italiano" agli occhi dei consumatori. La risposta a domande simili dovrebbe essere presa in considerazione prima di ricorrere alla tecnologia.

2.2.3 Diffusione della tecnologia nel settore agro-alimentare

Fino al 2015, la maggior parte degli articoli riguardanti la Blockchain trattava di criptovalute e crittografia. Nel 2015 , Noizat descriveva il primo tentativo di applicare la tecnologia a un nuovo campo (il voto elettronico) [23], e dal 2016 altri studi ne hanno approfondito l'applicazione come strumenti per la tracciabilità in varie filiere agroalimentari.

Dal 2016 diversi articoli hanno esplorato l'uso della blockchain come strumento per la tracciabilità nelle diverse filiere agroalimentari. Il numero di pubblicazioni totali ottenute attraverso il database di Web of Science, effettuando una ricerca che abbia come topic blockchain e food *(Topic = Blockchain AND Topic = Food)* è pari a 703 pubblicazioni.

La Tabella sottostante mostra quanto appena detto in coerenza con quanto già evidenziato nel paragrafo 2.1.6 del presente lavoro di tesi.

Anno	**Numero Pubblicazioni**	**% su 703**
2022	254	36.131%
2021	181	25.747%
2020	138	19.630%
2019	81	11.522%
2018	26	3.698%
2023	14	1.991%
2017	7	0.996%
2016	2	0.284%

Tabella 2.4: Numero di documenti con "Blockchain" e "Food" come topic su WebOfScience: suddivisione per anno.

2.3 Il settore vitivinicolo

Nel settore vitivinicolo, la gestione dei dati è estremamente importante, in quanto è una condizione necessaria per dimostrare al cliente le caratteristiche del prodotto e la sua qualità, garantendone la sicurezza, la tracciabilità e tutti i valori legati alla sostenibilità del prodotto e dell'organizzazione. In questo contesto, la raccolta

e la divulgazione dei dati diventano strumenti fondamentali di monitoraggio e comunicazione che consentono ai produttori di raccontare la storia dei loro marchi ai consumatori. Inoltre in Italia sono stati accertati numerosi casi di documentazione falsa nel settore vitivinicolo. Nel 2020, per esempio, con l'operazione "Dionisio" sono state arrestate cinque persone,principalmente in Lombardia ma sono state effettuate perquisizioni anche in Piemonte, Veneto, Emilia Romagna e Trentino Alto Adige, accusate di utilizzare fatture false per giustificare la falsificazione di indicazioni geografiche o denominazioni di origine (DOP e IGP) di prodotti agricoli, nonché di etichettare quantità di vino con denominazioni di pregio che in realtà non erano disponibili nella cantina. Al posto di questi vini, i produttori avrebbero usato vini di qualità inferiore rispetto a quanto dichiarato [24]. I vantaggi dell'utilizzo della blockchain per la tracciabilità del settore vitivinicolo sono chiari: i dati memorizzati dai sistemi blockchain sono irreversibili e trasparenti per tutte le parti interessate, il che li rende unici e fornisce credibilità all'intero sistema. Le informazioni contenute nel sistema blockchain consentiranno alle aziende di rafforzare il rapporto con i clienti attuali e di attrarne di nuovi condividendo processi e registrazioni [20] attraverso un'etichetta digitale posta su ogni bottiglia di vino. Per quanto riguarda i limiti, le scarse competenze informatiche interne rappresentano ancora uno dei maggiori limiti nell'adozione della tecnologia nel settore vitivinicolo, quindi la volontà di adottare e di investire in questa tecnologia dipendono anche dalla presenza di dipendenti con competenze informatiche all'interno dell'azienda.

2.3.1 Applicazioni già esistenti

In risposta a un'esigenza sempre più pressante e attenta alla sostenibilità dei prodotti, insieme alle possibilità di sviluppo e crescita del vino Made in Italy, sono state sviluppate e adottate dalle cantine italiane, tre app blockchain specializzate per il settore vinicolo: My StoryTM, Wine Blockchain EY e EY OpsChain [25].

- **My Story** [26], soluzione dell'ente di certificazione internazionale DNV GL, come suggerisce il nome, nasce per narrare l'intera storia della bottiglia di vino: garantisce l'autenticità delle bottiglie a fornitori, produttori e consumatori, rendendo i dati visibili e non manipolabili. Le informazioni raccolte sono accessibili inquadrando un QR-code oppure accedendo alla pagina web del sito principale. Le informazioni comprendono le caratteristiche del vitigno (luogo, terreno), i metodi di produzione, le certificazioni ottenute, le pratiche di trasformazione e le fasi di imbottigliamento. Attualmente, le cantine italiane che hanno fatto ricorso al sistema Mystory sono:

  1. Santella del Gröm Curtefranca Rosso DOC 2013 della cantina Ricci Curbastro in Franciacorta[10];
  2. Il Riserva Ducale Oro Chianti Classico Gran Selezione DOCG 2014 della cantina toscana Ruffino[11];
  3. Il Veritas Castel del Monte Bombino Nero Rosato DOCG 2017 della cantina pugliese Torrevento[12].

- **Wine Blockchain** [27], nasce da una partnership tra EY Italia e la startup EZ Lab. Grazie a questa collaborazione, la cantina Volpone[13], è stato il primo caso al mondo di cantina certificata blockchain. Wine Blockchain ha permesso la creazione di un registro pubblico legato alla firma digitale del produttore; i dati sono immutabili, mappano ogni passaggio della filiera, dal vigneto alla tavola e certificano l'origine delle materie prime, la loro territorialità, la qualità, l'autenticità e tutte le fasi della filiera del vino. Anche in questo caso, per

[10]https://mystory.dnvgl.com/product/f35bc07f-0bdf-440d-82f9-b367703bb01a/it/
[11]https://mystory.dnvgl.com/product/ca80bf57-b347-4da7-90da-d363421293f9/it/
[12]https://mystory.dnvgl.com/product/4aecf06d-85e2-4938-8638-d4f6d6b58a97/it/
[13]https://placidovolpone.it/blockchain-vini/

mezzo di un QR-code , il consumatore potrà verificare, in qualsiasi momento, la provenienza, le caratteristiche organolettiche e l'intera filiera agroalimentare e industriale della bottiglia certificate dal produttore.

- **EY OpsChain**, EY OpsChain [28], è stata utilizzata per creare la base della piattaforma di e-commerce del vino *TATTOO* (Traceability, Authenticity, Transparency, Trade, Origin, and Opinion) di Blockchain Wine Pte. Ltd. Questa piattaforma prevede token digitali che tracciano la provenienza, la qualità e l'autenticità dei vini nuovi e d'annata, eliminando i numerosi intermediari e consentendo un'efficienza dei costi; utilizza la soluzione blockchain di EY per promuovere e vendere vini in tutto il mondo, con particolare attenzione ai mercati della regione Asia-Pacifico, quindi Cina, Giappone, Corea del Sud, Thailandia e Singapore, dove il consumo di vini europei è in rapida crescita. EY OpsChain assegna inoltre un codice QR a ogni bottiglia che consente di accedere a informazioni sul produttore, sulla posizione del vigneto, sul campo in cui è stata coltivata l'uva, sui vitigni, sui pesticidi utilizza e sui trattamenti agricoli effettuati, sulle caratteristiche organolettiche, sulle condizioni di trasporto della partita per la trasformazione in vino e per la consegna e sul metodo di lavorazione. I vini sono gestiti sulla rete pubblica di Ethereum. Gli utenti hanno accesso a un'esperienza user-friendly implementata sulla soluzione SAP® Commerce. Il marketplace TATTOO Wine aiuta le aziende vinicole a caricare i cataloghi delle loro offerte e i consumatori a navigare tra le scelte di acquisto.

2.3.2 Il caso Placido Volpone

La cantina Placido Volpone, ubicata a Ordona, in provincia di Foggia, è stata la prima cantina al mondo a certificare la filiera del suo vino Falanghina su Blockchain. Tale azienda pugliese ha gentilmente fornito alcuni dati, successivi all'introduzione

della Blockchain, di seguito riportati, che hanno consentito di confermare come l'applicazione della stessa possa essere economicamente vantaggiosa per un'impressa vitivinicola. L'azienda *Placido Volpone*, come citato nel paragrafo precedente, ha fatto ricorso alla Wine Blockchain sviluppata da EZ Lab e EY. La tecnologia è stata introdotta nei processi aziendali nel 2017 e i dati riportati di seguito fanno riferimento al 2018:

Costo Fisso	40000€
Costo Variabile/Bottiglia	0,10€
Numero Bottiglie	130000
Costo Variabile Totale	13000€
Costo Totale Investimento	53000€

Tabella 2.5: Costi investimento Blockchain - Cantina Placido Volpone

Come sintetizzato nella Tabella 2.5, i costi totali sostenuti dall'azienda per ricorrere alla tecnologia sono stati di 53000€. Di questi, 40000€ rappresentano costi fissi, i restanti 130000€, invece, sono costi variabili.

Nello stesso periodo, la Cantina Volpone decide di applicare un *price premium* alle bottiglie di Falanghina di 2,20€. Di questo price premium, il 20% è associato all'introduzione della blockchain e l'80% al rebranding. Il prezzo della bottiglia passa dunque da 7€ a 9,20€: ricorrere alla tecnologia ha generato un beneficio di 57200€, generando così un surplus di 4.200€.

Prezzo/bottiglia pre-blockchain	7€
Prezzo/bottiglia post-blockchain	9,20€
Price Premium	2,20€
Beneficio	4200€

Tabella 2.6: Beneficio introduzione Blockchain - Cantina Placido Volpone

Si verifica inoltre, grazie alla digitalizzazione, ossia la radicale trasformazione del modo in cui vengono gestiti i dati e le informazioni, passando da un sistema cartaceo a uno completamente digitalizzato, una riduzione di costo pari a 3500€.

Quindi, il surplus totale ottenuto è di 7700€. Si può facilmente calcolare il ROI come Utile Netto/Capitale Investito (7700/53000) pari al 14,53%. Il risultato si mostra molto superiore rispetto alla media del settore dell'anno considerato che è di circa del 7% [29].

Surplus	7700€
Investimento Iniziale	53000€
ROI	14,53%

Tabella 2.7: ROI post-introduzione Blockchain - Cantina Placido Volpone

Il caso Placido Volpone conferma l'effetto positivo che può avere il ricorso alla tecnologia blockchain nella la creazione di valore per l'impresa. La soluzione adottata, oltre a ridurre i costi burocratici, ha permesso di rendere i consumatori consapevoli della qualità, sicurezza e sostenibilità dei prodotti, in quanto consente di tracciarne il ciclo di vita: il consumatore viene reso partecipe dell'intero processo produttivo.

Altri risultati importanti sono quelli della *brand awereness* e della *brand recognition*: se qualche anno fa l'azienda era nota a livello locale, oggi grazie all'introduzione della blockchain che ha comportato innovazione e trasparenza, si è affermata a livello mondiale.

2.4 Cos'è una DApp

Nuovi modelli informatici emorgono ogni 10-15 anni: i PC alla fine degli anni '70, Internet all'inizio degli anni '90 e gli smartphone alla fine degli anni 2000. Ogni modello informatico ha permesso di creare nuove classi di applicazioni che hanno sfruttato i punti di forza unici della piattaforma. Ad esempio, gli smartphone sono stati i primi veri personal computer ad integrare sensori come il GPS e le fotocamere ad alta risoluzione e applicazioni come Instagram, sfruttando queste capità uniche, vengono utilizzate da miliardi di persone [30]. La Blockchain è stata proposta per la prima volta nel 2008 e a sfruttare le caratteristiche di questa piattaforma sono le *applicazioni decentralizzate (dApp)*. Una DApp [31] è un'applicazione web "abilitata alla catena di blocchi" che gira su una rete di computer peer-to-peer e non su un singolo server. L'origine delle applicazioni decentralizzate risale all'avvento delle reti peer-to-peer, architetture distribuite che suddividono compiti o carichi di lavoro tra peer. BitTorrent[14] è un esempio di applicazione decentralizzate che gira su reti peer-to-peer. Una dApp include sia il front-end che il back-end e funziona in modo indipendente su tutti i nodi. La differenza fondamentale con un'applicazione tradizionale è uno smart contract che collega l'applicazione a una rete blockchain.

[14]programma client di distribuzione e condivisione di file di tipo peer-to-peer

2.4.1 Perchè una DApp per il settore vitivinicolo

Dall'analisi svolta nel presente Capitolo, è emerso un crescente interesse per la Blockchain, confermato dalle sempre più frequenti e numerose pubblicazioni scientifiche sull'argomento.

La funzione più richiesta alla tecnologia è quella di coordinare e tracciare le informazioni, caratteristica che ne consente la piena applicazione nel settore agroalimentare. In particolare, la realizzazione di una DApp rende i consumatori partecipi del processo produttivo e risponde alla loro crescente richiesta di trasparenza. Infatti, dai risultati delle indagini svolte è emerso che i consumatori finali sono disposti a pagare un prezzo più elevato pur di assicurarsi qualità del prodotto e sostenibilità dei processi produttivi.

Infine, come conferma il caso Placido Volpone, ricorrere alla blockchain costituisce un vantaggio per le PMI: oltre a garantire un ROI più elevato della media del settore, consentirà all'impresa di differenziarsi dai suoi competitor.

Capitolo 3

Analisi dei requisiti

L'analisi dei requisiti rappresenta il processo di raccolta, definizione e organizzazione dei requisiti di un sistema o di un progetto. Consiste nell'identificare le esigenze degli stakeholder (utenti, clienti, proprietari del sistema, etc.) e nello stabilire i requisiti del sistema tali da soddisfare le esigenze individuate.

3.1 Descrizione del caso studio

Dimostrata la validità dell'applicazione della tecnologia blockchain al settore agroalimentare, e, in particolare, a quello vitivinicolo, il presente lavoro di tesi si propone di realizzare un prototipo di applicazione decentralizzata basata su Ethereum e pensata per un'azienda vitivinicola. L'aumento delle contraffazioni, delle frodi e dell'uso eccessivo di conservanti,ha reso necessario tracciare la filiera vitivinicola in modo da permettere al consumatore di verificare la qualità e sostenibilità di ogni bottiglia di vino. Questo avrà un effetto positivo per l'azienda che vedrà aumentare la visibilità del brand.

La maggior parte dei sistemi attuali sono basati su pagine web, è quindi possibile contraffare le informazioni memorizzate. Il prototipo realizzato, basandosi su

Ethereum, invece, garantisce informazioni immutabili: propone un sistema per dimostrare l'autenticità del prodotto, registrando informazioni dettagliate in un database immutabile e incorruttibile. Qualsiasi cambiamento delle informazioni registrate interromperà la catena [32].

Questo progetto è stato pensato per una PMI del novarese. Il Podere ai Valloni si colloca nel comune di Boca, in provincia di Novara ed è impegnato nella produzione di vino e, in particolare, di Boca DOC di altissima qualità; tanto che, quello ottenuto dalle uve di *Vigna Cristiana*, è stato inserito tra i vini di ecccellenza italiani dalla *Slow Food*[1] nella Banca del Vino.

L'azienda vuole ricorrere alla tecnologia blockchain per garantire la trasparenza, la qualità e la territorialità del proprio vino, per rispondere alle esigenze di una domanda sempre più attenta alle tematiche ambientali e per migliorare la comunicazione con gli stakeholder.

3.2 Requirements Engineering Process

L'ingegneria dei requisiti è l'area dell'ingegneria dei sistemi[2] che si occupa del processo di sviluppo e di verifica dei requisiti che deve implementare il sistema. Seguire buone pratiche di ingegneria dei requisiti aiuta a raggiungere l'obiettivo primario, ossia che il sistema consegnato soddisfi le esigenze del cliente [33].

Lo scopo del Requirements Engineering è di aiutare a comprendere cosa realizzare prima che inizi lo sviluppo del sistema in modo da prevenire il ricalcolo dei costi, in quanto, più tardi saranno compresi gli errori, più sarà economicamente dispendioso

[1] https://www.slowfood.it/

[2] branca interdisciplinare dell'ingegneria che si occupa dello sviluppo e organizzazione di sistemi artificiali complessi

correggerli. É possibile minimizzare il numero di errori definendo un insieme stabile di requisiti prima di iniziare la progettazione e l'implementazione del sistema [34].

Gli step adoperati nel RE process sono quelli identificati nel paper "Understanding the Requirement Engineering for Organization:The Challenges" [34]:

- *requirements elicitation*,
- *requirements analysis & negotiation*,
- *requirements documentation*,
- *requirements verification & validation*,
- *requirements management*.

3.2.1 Requirements Elicitation

Il processo di ingegneria dei requisiti inizia con la raccolta dei requisiti che avviene tramite un processo di elicitazione: è un processo di ricerca, scoperta, acquisizione ed elaborazione dei requisiti per lo sviluppo del sistema. Esso consiste nella definizione della visione del prodotto e dell'obiettivo di progetto e nell'identificazione di stakeholder, clienti e users.

Gli stakeholder individuati sono:

- *produttore*, è l'azienda vitivinicola che utilizza il sistema per fornire un servizio di trasparenza ai suoi consumatori;
- *fornitori*, i produttori di tappi e botti;
- *consumatore finale*, il cliente finale che acquista il prodotto dal produttore stesso o presso un punto di rivendita.

Le tecniche adoperate per questa prima fase sono state interviste e brainstorming.

Il processo ha avuto inizio con un'intervista all'azienda cliente grazie alla quale sono stati definiti gli aspetti principali a cui deve rispondere il prodotto. Sono state poste domande specifiche per capire le reali esigenze dell'azienda e le sue aspettative relativamente al funzionamento del prodotto, successivamente sono stati richiesti dei feedback su possibili opzioni di progettazione.

Vengono di seguito sintetizzate le domande poste e le relative risposte ottenute.

- Il primo punto affrontato è stato decidere cosa memorizzare su Blockchain, se il lotto o la singola bottiglia. Il Podere ai Valloni presenta una produzione piuttosto limitata, circa 12000 bottiglie all'anno suddivise su tre tipologie di vini. Gli imbottigliamenti avvengono per monoblocco su tipologia quindi, in una giornata, l'azienda imbottiglia un'annata. Di conseguenza, il prodotto memorizzato identificherà il lotto di bottiglie per annata e tipologia di vino. Non viene presa in considerazione la possibilità di svolgere tale lavoro per singola bottiglia perché richiederebbe un elevato impiego di risorse economiche, significativo solo per vini molto costosi.

- Il vino più pregiato dell'azienda è il *Boca DOC Vigna Cristiana* ed è quello a cui si vuole dare maggiore visibilità. Viene prodotto da quarant'anni e, già con la sua denonimazione, "Boca", richiama il suo stretto legame con il territorio. Inizialmente l'azienda voleva applicare la tecnologia solo per il Boca DOC; con il procedere dell'intervista, però, ci si è resi conto che si correva il rischio di trascurare le altre due tipologie di vino, che potevano essere percepiti dal consumatore come prodotti di inferiore valore e qualità. Di conseguenza, si è deciso di consentire anche l'inserimento di altre tipologie di vino.

- Per quanto riguarda la *tipologia di dati da memorizzare*,dall'intervista è emerso che le informazioni su cui si vuole dare massima trasparenza, le più rilevanti

del settore vitivinicolo tanto da determinare le caratteristiche del vino, sono quelle relative all'ubicazione geografica del vigneto e le caratteristiche del suolo. Viene richiesto inoltre il salvataggio di informazioni relative alle caratteristiche della singola bottiglia (composizione, caratteristiche chimiche e organolettiche etc). Necessario è anche l'inserimento del processo di produzione. Si vogliono inoltre salvare foto, video e documenti. Questi ultimi serviranno soprattutto per dimostrare l'autenticità del prodotto italiano all'estero. I documenti a disposizione sono:

1. certificato di idoneità di denominazione di origine controllata rilasciato da AgroQualità[3];
2. certificazione etica ed ambientale, a testimonianza di un prodotto biologico rilasciata da ICEA[4].

- Inoltre, l'azienda richiede una pagina di visualizzazione per il consumatore: è necessario creare una pagina web dove il consumatore possa accedere alle informazioni salvate in Blockchain. Tale pagina deve essere accessibile per mezzo di un QR-code. Relativamente a questo, però, è emerso un problema: sulla bottiglia al momento è presente un solo QR-code ma, nell'immediato futuro ne sono previsti altri due. I QR-code da inserire obbligatoriamente sulla bottiglia saranno quelli relativi a:
  - enoturismo;
  - etichetta nutrizionale;
  - bilancio di sostenibilità.

[3]https://www.rina.org/it/agroqualita

[4]https://icea.bio/

Aggiungere un quarto QR-code, ossia quello relativo alla blockchain, potrebbe creare confusione al consumatore. É stata dunque proposta come soluzione la creazione di un unico QR-code (quello della blockchain) che includa nella pagina web anche i link relativi agli altri tre. Un'alternativa potrebbe essere quella di salvare su blockchain anche le stringhe dei link associati alla bottiglia.

É importante sottolineare che la cantina è di proprietà dell'azienda al 100%, tutte le fasi necessarie per la realizzazione del vino sono svolte internamente e per questo non viene sfruttata un'ulteriore potenzialità della Blockchain: se l'azienda avesse avuto rapporti con dei fornitori di uva, utilizzando la Blockchain, avrebbe avuto sia la certificazione, sia la determinazione di responsabilità nei confronti dei fornitori. I prodotti forniti da aziende esterne sono solo tappi e botti.

Per questa prima fase dimostrativa, alcune possibili funzionalità sono state trascurate in quanto l'azienda non può al momento fornire un'adeguata documentazione. Infatti, potrebbe essere utile registrare su Blockchain l'avvio della produzione, la durata di ogni fase produttiva, relative immagini esplicative, la manutenzione delle attrezzature e altri dati correlati. La registrazione di questi dati su Blockchain consentirebbe di avere un registro permanente e immutabile di tutte le attività produttive, garantendo una maggiore tracciabilità e trasparenza nel processo produttivo. Inoltre, il registro su blockchain potrebbe anche essere utile per l'identificazione rapida di eventuali problemi o inefficienze nella produzione e per migliorare la qualità del prodotto finale.

A tal proposito, è importante evidenziare che il prototipo sviluppato potrà essere implementato in futuro: rappresenta solo una versione preliminare del prodotto finale, sviluppata principalmente per dimostrare la fattibilità del concetto e ottenere un feedback iniziale dal cliente. Come tale, non comprende tutte le funzionalità e le caratteristiche richieste dalla specifica ma queste possono essere implementate in una fase successiva dello sviluppo.

3.2.2 Requirements Analysis & Negotiation

Nella seconda fase del presente lavoro, i requisiti identificati durante il processo di raccolta, vengono integrati e analizzati.

I *requisiti* sono una specifica di ciò che dovrebbe essere implementato; sono descrizioni di come dovrebbe funzionare il sistema,una sua proprietà o un suo attributo. I requisiti software sono divisi in:

- *requisiti funzionali*, direttamente correlati a ciò che il sistema deve fare;
- *requisiti non funzionali*, rappresentano delle caratteristiche che, pur non essendo funzionalità, dovranno essere soddisfatte dalla piattaforma.

Requisiti funzionali del sistema

Il sistema deve permettere al produttore di:

- aggiungere nuovi prodotti e le relative caratteristiche;
- aggiungere tutte le caratteristiche relative al contesto geografico;
- aggiungere una descrizione del processo di produzione con foto e video;
- salvare documenti e certificazioni;
- poter navigare tra i prodotti salvati.

Il sistema deve permettere al cliente di:

- accedere alla visualizzazione dei dettagli del prodotto per mezzo di un QR-code;
- visulaizzare i link associati ai qr-code obbligatori sull'etichetta.

Requisiti non funzionali del sistema

Il sistema deve:

- essere basato sulla tecnologia blockchain per garantire la trasparenza e l'immutabilità dei dati. La piattaforma dovrà essere dunque completamente decentralizzata, la decentralizzazione dovrà riguardare anche i file caricati dall'utente;
- essere facile da utilizzare per il produttore, dovrà presentare un'interfaccia user-friendly;
- presentare una pagina di visualizzazione esplicativa per il consumatore;
- garantire che il costo di utilizzo e di sviluppo si contenuto.

3.2.3 Requirements Documentation

La fase di Requirements Documentation si riferisce al processo di documentazione dei requisiti: l'obiettivo è la comunicazione dei requisiti tra le parti interessate. Lo sviluppo del progetto parte da opportuni modelli astratti e semplificati. La modellazione è stata svolta adottando lo Unified Modeling Language (UML) che è un linguaggio di modellazione, ovvero un insieme di costrutti e regole a cui è associata una semantica ben definita [35].

In questa fase sono stati adottati:

- ***diagramma delle classi UML*** (Class Diagram), per la modellazione concettuale;
- ***diagramma dei casi d'uso UML*** (Use Case Diagram), per la modellazione funzionale.

Questi due elementi, oltre a fornire una solida base di partenza per l'implementazione del progetto, forniscono una visione schematica e permettono di avere una visione d'insieme più chiara.

Diagramma delle classi UML

La modellazione concettuale consiste nel costruire un modello che fornisca una descrizione ottimale delle informazioni che vengono acquisite, manipolate e memorizzate dal sistema Il diagramma delle classi UML [36], riportato in Figura 3.1, è stato impiegato al fine di schematizzare i concetti e gli elementi che caratterizzano il sistema.

La classe principale è la classe *Prodotto* e corrisponde al contratto sviluppato. L'attributo chiave del prodotto è l'id che lo identifica. Le classi Dettagli, Contesto e Processo sono definite come struct nello smart contract quindi definite con relazioni uno a uno alla classe principale, in quanto, ad un prodotto deve essere necessariamente associata la struct così come la struct deve essere necessariamente associata a un prodotto. Nel diagramma delle classi sono state inoltre definite due classi fittizie (consumatore e produttore) per semplificare la rappresentazione concettuale, ma queste classi non sono state implementate o sviluppate ulteriormente; questo approccio consente di rendere più semplice la comprensione del sistema agli stakeholder. Queste però potrebbero essere implementate in una fase successiva dello sviluppo del software, quando si ha bisogno di ulteriori dettagli sulla loro funzionalità.

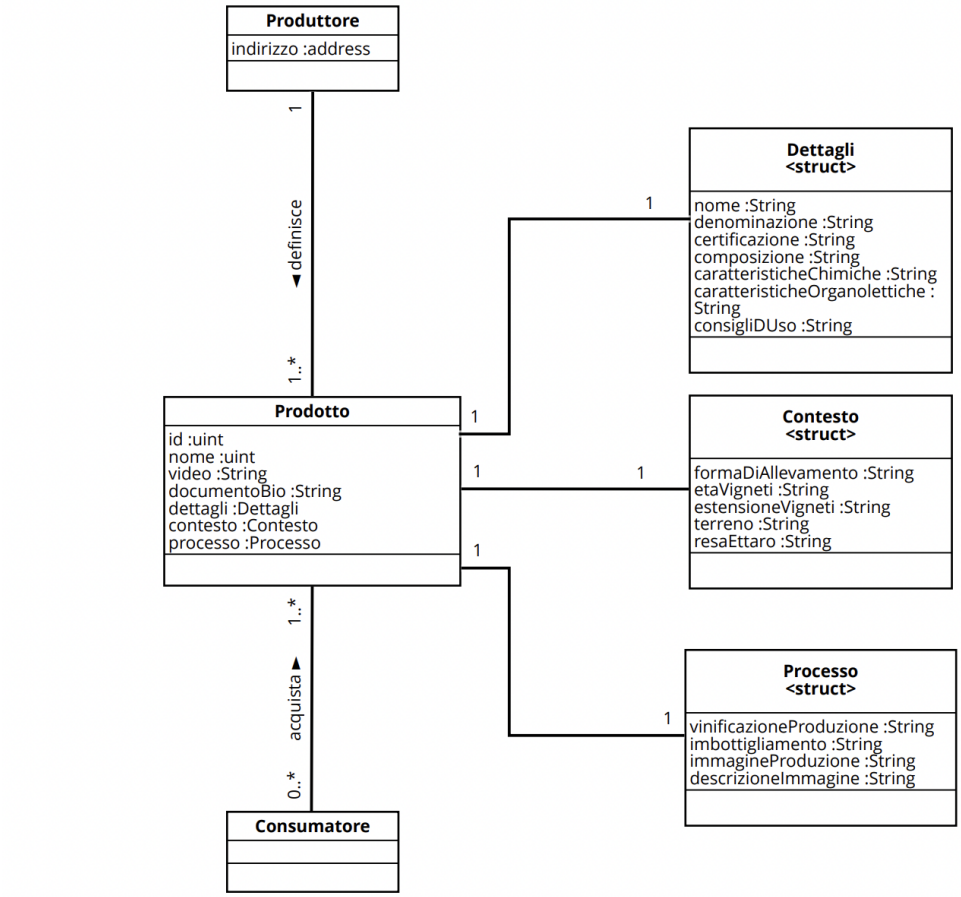

Figura 3.1: Class Diagram

Diagramma dei casi d'uso UML

Gli elementi del diagramma delle classi possono essere legati tra loro da relazioni. La più comune è quella di partecipazione che lega un attore al caso d'uso a cui partecipa. L'obiettivo dei diagrammi dei casi d'uso è di fornire una visione d'insieme del sistema in termini di attori che interagiscono con esso e dei loro obiettivi. Si ottiene quindi una descrizione di alto livello, priva di dettagli, che indica quali sono le finalità senza fornire ulteriori informazioni su come queste possano essere

raggiunte tramite interazioni con il sistema [35].

Gli attori coinvolti nel flusso di informazioni sono:

- produttore, è l'attore principale del flusso, inserisce nuovi prodotti nel sistema;
- consumatore, visualizza informazioni sul prodotto registrate sulla blockchain.

Il diagramma dei casi d'uso UML descrive i requisiti funzionali del sistema. Nel caso in esame è stata effettuata l'analisi rispetto al livello Summary che è il livello più generale e ampio e permette di raggruppare più casi d'uso di livello user-goal in un insieme di obiettivi tra di loro correlati.

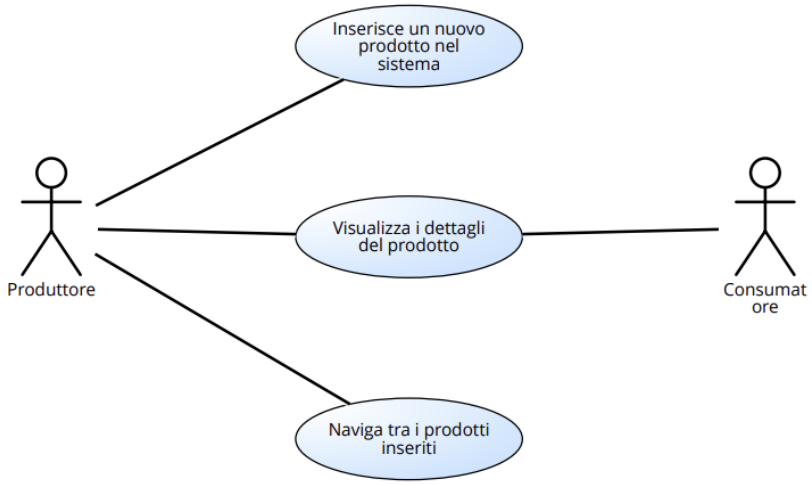

Figura 3.2: Use Case Diagram.

3.2.4 Requirements Verification & Validation

Una volta specificati i requisiti, è importante controllare che siano completi, coerenti e soddisfacenti per tutti gli stakeholder secondo i seguenti step:

1. verificare che i requisiti raccolti siano completi e coerenti;
2. verificare che siano soddisfacenti per tutti gli stakeholder (produttore, fornitore e cliente);

3. valutare la fattibilità tecnica e economica del sistema, assicurandosi che sia possibile sviluppare e implementare il sistema nel rispetto dei tempi e dei budget stabiliti.

Questo aspetto verrà abbondantemente discusso nel Capitolo successivo.

3.2.5 Requirements Management

Dopo che i requisiti sono stati specificati e validati, è importante gestirli durante tutto il ciclo di vita del progetto, assicurandosi che siano soddisfatti e che il sistema soddisfi le esigenze dei stakeholder.

Capitolo 4

Tecnologie utilizzate

Terminato il lavoro di definizione dei requisiti, sono state individuate le tecnologie e gli strumenti da utilizzare in fase di implementazione.

4.1 Scelta della piattaforma

Per la realizzazione della dApp sono state valutate due piattaforme Blockchain: Ethereum e Hyperleger. Le Blockchain di prima generazione erano concepite per supportare le criptovalute; fornivano una capacità limitata di supportare qualsiasi logica commerciale, escludendo il trasferimento di valore tra conti. La seconda generazione, invece, (ad esempio, Ethereum) è stata concepita come un'infrastruttura programmabile peer-to-peer di uso generale, in grado di supportare programmi complessi noti come smart contract [10]. Ethereum e Hyperledger sono entrambe piattaforme di blockchain, ma con scopi e caratteristiche differenti.

Ethereum è una piattaforma open-source decentralizzata che permette di creare applicazioni decentralizzate (DApps) basate su blockchain. Utilizza Solidity come linguaggio di programmazione per creare smart contract ed è noto anche per la

sua criptovaluta nativa, l'Ethereum (ETH), che viene utilizzata per pagare le transazioni sulla rete.

Hyperledger è un progetto open-source ospitato dalla Linux Foundation[1] che sviluppa tecnologie blockchain aziendali. Hyperledger si concentra su fornire un'infrastruttura di blockchain per aziende e organizzazioni che desiderano creare soluzioni blockchain per gestire la catena di approvvigionamento, il tracciamento delle transazioni finanziarie e i dati dei clienti. L'obiettivo di Hyperledger è supportare sistemi blockchain e registri distribuiti efficienti e stabili. Offre diversi framework blockchain, tra cui Hyperledger Fabric, Hyperledger Sawtooth e Hyperledger Burrow, che possono essere utilizzati per creare applicazioni blockchain personalizzate. In particolare per il presente lavoro è stato valutato l'utilizzo di Hyperledger Fabric che permette lo sviluppo di dApp e di Smart Contract. Come i contratti intelligenti, Hyperledger fabric consente anche alle organizzazioni aderenti di eseguire alcuni codici sui peer che creano le transazioni a una condizione specifica. Questi codici sono noti come chaincode.

Il primo step adottato per la scelta della piattaforma è stato capire come si stesse muovendo il mondo delle start-up in tale ambito. Quello che si è evinto grazie allo studio di Nikhil Vadgamam e Paolo Tasca [17], che approfondisce la valutazione delle blockchain utilizzate dai progetti in base al tipo di organizzazione,è che Ethereum domina la categoria delle startup, con il 32% di tutti i progetti di startup che utilizzano questa piattaforma. Per i consorzi e le società pubbliche, Hyperledger è il più utilizzato rispettivamente con il 35% e il 34%. Per le iniziative del governo, il 50% dei progetti non ha identificato le blockchain utilizzate (TBC). Di tutti i progetti Ethereum, il 90% è stato utilizzato dalle startup. Di tutti i progetti Hyperledger, consorzi, società pubbliche e startup hanno utilizzato questa

[1]https://www.linuxfoundation.org/

blockchain al 24%, 28% e 45%.

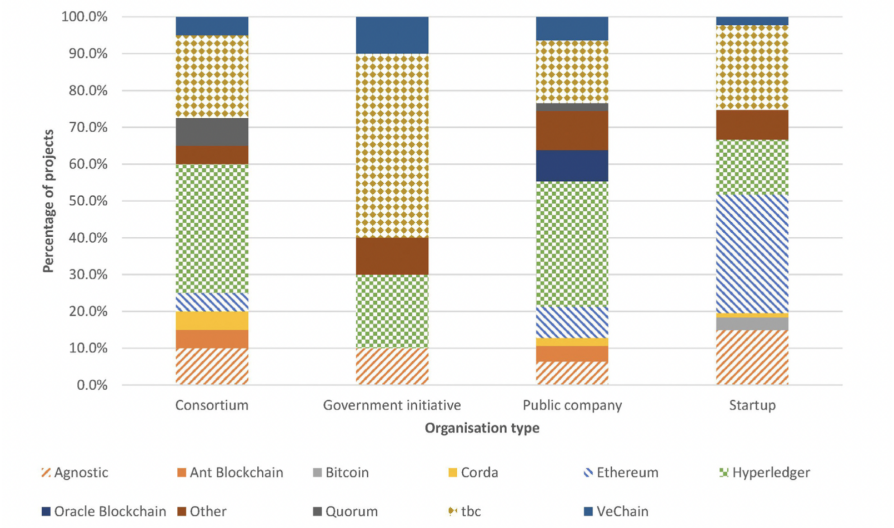

Figura 4.1: Ethereum vs Hyperledger - Startup

Inoltre, sono stati valutati i costi delle piattaforme, le richieste della start-up e dell'azienda vitivinicola.

Per quanto riguarda i costi, Ethereum è una piattaforma gratuita, non ci sono costi diretti associati all'uso della stessa. Tuttavia, ci sono costi di transazione associati all'utilizzo della rete per eseguire smart contract o inviare criptovaluta ETH. Hyperledger, d'altra parte, è un progetto open-source che richiede l'implementazione e la personalizzazione da parte delle aziende o delle organizzazioni, il che potrebbe comportare costi di sviluppo e di implementazione, generando costi associati alla gestione e alla manutenzione della rete Hyperledger personalizzata.

Dunque, il potente motore di contratti intelligenti di Ethereum lo rende una piattaforma generica per qualsiasi tipo di applicazione. Tuttavia, la sua modalità di funzionamento senza permessi e la sua totale trasparenza vanno a discapito della scalabilità delle prestazioni e della privacy. Hyperledger, d'altro canto, risolve i problemi di scalabilità delle prestazioni e di privacy grazie alla modalità di funzionamento permissioned e al controllo degli accessi. Inoltre, l'architettura

modulare consente a Hyperledger di essere personalizzato per una moltitudine di applicazioni.

In definitiva, si è optato per Ethereum in quanto si presenta come la scelta vincente quando le organizzazioni in via di sviluppo intendono creare applicazioni decentralizzate per l'uso da parte dei clienti. Conviene ricorrere a Hyperledger, invece, quando ci sono dati sensibili da salvare, offrendo la possibilità di creare un'applicazione blockchain mantenendo la privacy delle informazioni dell'organizzazione oppure quando un'organizzazione o un'azienda vuole definire algoritmi blockchain propri e unici. Infatti, nei progetti Hyperledger, l'intera infrastruttura sottostante della blockchain può essere modificata. Questa flessibilità si rivela un ottimo strumento per la creazione di applicazioni blockchain personalizzate per scopi aziendali.

4.1.1 Solidity

Solidity è un linguaggio di programmazione ad alto livello orientato agli oggetti, impiegato per l'implementazione di contratti intelligenti eseguiti su Ethereum. Solidity, influenzato da noti linguaggi di programmazione tra cui C++, Python e JavaScript, è stato pensato per la Ethereum Virtual Machine (EVM) [37]. Le caratteristiche più importanti di Solidity sono: la tipizzazione statica[2], ereditarietà, librerie e tipi complessi definiti dall'utente. In particolare si tratta di un linguaggio di programmazione fortemente tipizzato (il programmatore ha la possibilità di indicare alla macchina quale tipo di dati aspettarsi), a differenza dei linguaggi a tipizzazione libera che non richiedono al programmatore di essere specifico nella definizione del tipo di dato. Solidity offre la possibilità di creare dei loop, presenta infatti le stesse regolarità sintattiche di JavaScript e C quando si tratta di loop [38].

[2]In programmazione, la tipizzazione statica consiste nell'assegnazione di tipi alle variabili.

La versione di Solidity utilizzata è stata dalla 0.8.7 in su.

4.1.2 Smart Contract

Il contratto intelligente, scritto in linguaggio Solidity, è l'unità di funzionalità che viene caricata sulla macchina virtuale di Ethereum. Comprende sia la parte di codice che di dati, non è mai controllato da un utente ma opera sempre secondo la programmazione. Per questo motivo, il protocollo è in grado di processare le informazioni di per sé e in maniera infallibile.

A livello di sorgente, i contratti appaiono simili alle classi dei linguaggi orientati agli oggetti. Questi possono contenere dichiarazioni di variabili di stato (analoghe ai campi della classe), definizioni di funzioni (metodi), modificatori, costruttori e strutture, supportano l'incapsulamento (attributi di visibilità) e possono anche ereditare da più contratti [39].

Una caratteristica importante degli smart contract è la componibilità che, in generale, consiste nel combinare componenti distinti per creare nuovi sistemi o prodotti. Quindi, gli sviluppatori possono riutilizzare i componenti software esistenti per creare nuove applicazioni: chiunque può interagire con il contratto o integrarlo in una dApp per aggiungere funzionalità.

La componibilità degli smart contract si basa su tre principi:

- *modularità*: la capacità dei singoli componenti di svolgere un compito specifico. In Ethereum, ogni smart contract ha un caso d'uso specifico;
- *autonomia*: i componenti componibili devono essere in grado di operare in modo indipendente. In Ethereum ogni smart contract è autoesecutivo e può funzionare senza dipendere da altre parti del sistema;
- *scopribilità*: gli sviluppatori non possono chiamare contratti esterni o integrare librerie software nelle applicazioni se i primi non sono pubblicamente disponibili.

I contratti intelligenti sono open-source: chiunque può richiamare un contratto intelligente o eseguire il fork di una base di codice.

Tutti gli smart concract prevedono una sequenza di condizioni che devono essere rispettate e, se le condizioni specifiche vengono confermate, i dati sono disponibili sulla blockchain. L'effetto principale dell'esecuzione del codice del contratto smart è l'alterazione dello stato della rete Ethereum.

4.2 Node.js

Node.js [40] è un runtime JavaScript *asincrono event-driven* progettato per costruire applicazioni di rete scalabili. JavaScript è uno dei linguaggi di programmazione più diffusi al mondo, inizialmente utilizzato principalmente per lo scripting lato client ma disponibile solo all'interno del tag <script> dell'HTML di una pagina web, costringeva gli sviluppatori ad utilizzare più linguaggi e framework per lavorare contemporaneamente tra componenti front-end e back-end. Node.js costituisce dunque un ambiente di runtime che contiene tutto il necessario per eseguire programmi scritti in JavaScript.

4.3 Truffle

Per lo sviluppo di smart contract è stato utilizzato il framework *Truffle* [41], in quanto, quest'ultimo, fornisce una serie di strumenti per scrivere smart contracts in Solidity, effettuare test con Mocha[3] e Chai[4] (librerie di testing per Node.js) e distribuire contratti sulla blockchain che utilizzano la macchina virtuale di Ethereum (EVM). Tra i vari comandi di Truffle i più utilizzati sono stati:

[3]https://mochajs.org/

[4]https://mochajs.org/

1. *truffle compile*, digitando il comando nel terminale nella directory in cui si trova il progetto, tutti i contratti vengono compilati. Nelle esecuzioni successive, Truffle compilerà solo i contratti che sono stati modificati.

2. *truffle migrate*, questo comando eseguirà tutte le migrazioni presenti nella cartella *migrations* del progetto. Le migrazioni sono file JavaScript necessari per distribuire i contratti attraverso la rete Ethereum. La cronologia delle migrazioni precedentemente eseguite viene registrata on-chain tramite un contratto di migrazione (Migration.js). Per i test locali, è stata necessaria una blockchain di prova come Ganache configurata e funzionante prima di eseguire il comando.

4.3.1 Ganache

Per testare i contratti intelligenti è stata utilizzata Ganache [42], una blockchain privata per il rapido sviluppo di applicazioni decentralizzate su Ethereum e Corda. Ganache può essere utilizzata durante tutto il ciclo di sviluppo, consentendo l'implementazione e il test delle dApp in un ambiente sicuro.

La configurazione di default di Ganache prevede le caratteristiche riportate di seguito.

```
Hostname: 127.0.0.1 - localhost
Port Number: 7545
Network ID: 5777
Automine: true
Error on Tx Failure: true

Account Default Balance: 100
Total Accounts to Generate: 10
Autogenerate HD Mnemonic: false
Lock Accounts: false
```

Figura 4.2: Configuarazione di Ganache.

Nella Figura 4.2, *automine:true* indica che Ganache inizia automaticamente a minare blocchi dopo ogni transazione. Ganache prevede 10 account con un balance di 100 ETH ciascuno. Inoltre, il Gas Limit (quantità massima di gas disponibile per ogni blocco e transazione) e il Gas Price (il prezzo di ogni unità di gas, in WEI) sono rispettivamente di 6721975 e 20000000000 wei. Il carburante (gas) si riferisce alla commissione richiesta per effettuare una transazione su Ethereum. Questa commissione viene pagata nella valuta nativa di Ethereum: l'Ether (ETH). I prezzi del carburante sono visualizzati in Gwei dove, il wei è l'unità più piccola di ETH. Un Gwei, termine che sta per "giga-wei" equivale a 0,000000001 ETH (10-9 ETH).

Per testare gli smart contract è stata utilizzata Ganache perchè offre i seguenti vantaggi:

- **velocità**: consente di testare e convalidare rapidamente le funzionalità della dApp, poiché le transazioni vengono elaborate istantaneamente sulla rete locale;

- **sicurezza**: l'utilizzo di una rete locale per lo sviluppo evita l'esposizione ad attacchi esterni e malfunzionamenti della rete principale;

- **flessibilità**: offre molte opzioni di configurazione, inclusa la possibilità di emulare vari nodi di rete, configurare le velocità di elaborazione delle transazioni e controllare le impostazioni di rete;

- **semplicità**: è facile da utilizzare, semplifica l'integrazione di applicazioni blockchain con altri servizi e piattaforme in quanto è compatibile con popolari strumenti di sviluppo blockchain, come Truffle e Remix.

4.4 Web3.js

Una DApp si compone di back-end e front-end e l'interazione tra le due parti è possibile grazie alla libreria web3.js. Web3.js [43] è una raccolta di librerie che permette l'interazione con un nodo Ethereum remoto o locale utilizzando HTTP, IPC o WebSocket. Grazie a Web3.js, è possibile sviluppare siti web o client che interagiscono con la blockchain. Si tratta di azioni come l'invio di Ether da un utente a un altro, la verifica dei dati dei contratti intelligenti, la creazione di contratti intelligenti e altro ancora. Web3.js comunica con Ethereum Blockchain con il protocollo JSON RPC (Remote Procedure Call) permettendo di fare richieste a un singolo nodo Ethereum per leggere e scrivere dati sulla rete. La libreria Web3.js permette di sfruttare nel browser la maggior parte delle funzionalità back-end fornite dalla rete Ethereum.

4.5 IPFS

L'Interplanetary File System (IPFS) [44] è una rete distribuita che consente l'archiviazione e l'accesso a contenuti di ogni tipo e dimensione: file, siti web, applicazioni e dati. IPFS è una rete peer-to-peer in cui gli utenti memorizzano e accedono ai contenuti direttamente da altri utenti della rete piuttosto che da un server centrale.

I vantaggi di ricorrere a IPFS sono:

- *decentramento*: poiché IPFS è una rete peer-to-peer, i contenuti possono essere archiviati e condivisi senza fare affidamento su un server centrale, rendendolo più resistente alle interruzioni;

- *sicurezza*: il contenuto IPFS è identificato da un hash crittografico che ne garantisce l'autenticità e la resistenza alla manomissione, questo lo rende adatto per la distribuzione e l'archiviazione di contenuti;

- *efficienza*: può ridurre la duplicazione dei contenuti archiviando solo contenuti univoci e riutilizzandoli in rete.

Per quanto concerne il suo funzionamento, indipendentemente dal fatto che il file sia archiviato sul nodo locale o su un nodo gestito da un servizio o un'applicazione di pinning compatibile con IPFS (come Pinata), quando si aggiunge un file a IPFS, il file viene suddiviso in parti più piccole, sottoposto a hash crittografico e dotato di un'impronta digitale univoca chiamata *Content Identifier* (CID). Il CID funge da registrazione permanente del file così come esiste in quel momento. Quando altri nodi cercano un file, chiedono ai nodi peer che memorizzano il contenuto a cui fa riferimento il CID del file. I nodi possono bloccare il contenuto per conservarlo per sempre (e renderlo disponibile) o scaricarlo per risparmiare spazio. Ciò significa che ogni nodo della rete memorizza solo il contenuto di interesse e alcune informazioni di indicizzazione. Quando viene aggiunta una nuova versione di un file a IPFS, viene generato un nuovo CID perché il suo hash crittografico è diverso. Ciò significa che i file archiviati su IPFS sono resistenti alla manipolazione e alla censura. La modifica di un file non sovrascrive l'originale e i punti in comune tra i file possono essere riutilizzati per ridurre al minimo i costi di archiviazione.

4.5.1 Pinata

Pinata [45] è un servizio di gestione dei media NFT che consente agli utenti di ospitare, gestire e condividere file di qualsiasi tipo sulla blockchain; è il modo più semplice per inserire contenuti in IPFS e creare applicazioni web3 senza dover costruire e gestire i propri nodi IPFS.

Si è optato per Pinata per la sua facilità d'uso: semplifica enormemente il processo di caricamento e gestione dei file su reti IPFS fornendo un'interfaccia utente intuitiva e strumenti di gestione avanzati. Offre funzionalità avanzate di

memorizzazione nella cache e una rete CDN [5] globale per un accesso ai dati rapido e affidabile ovunque. Per quanto concerne la sicurezza, Pinata offre crittografia end-to-end e il controllo degli accessi così da garantire che i dati siano sempre al sicuro. In definitiva, per il presente progetto, Pinata risulta la scelta migliore per la gestione dei file su reti IPFS in modo semplice e affidabile. L'utilizzo diretto di IPFS può essere difficoltoso in quanto richiede la conoscenza di molte funzionalità tecniche e complesse: la configurazione di IPFS, infatti, richiede l'installazione di una serie di pacchetti software e uno studio approfondito di problematiche specifiche come la gestione dei protocolli di rete P2P e le modalità di archiviazione dei dati. Con IPFS la disponibilità dei dati può essere influenzata dalla disponibilità dei nodi di rete e richiede di lavorare da linea di comando. Al contrario, Pinata fornisce un'interfaccia grafica intuitiva e user-friendly che consente agli utenti di gestire facilmente il caricamento e la distribuzione dei dati tramite IPFS senza necessariamente conoscere i dettagli tecnici sottostanti.

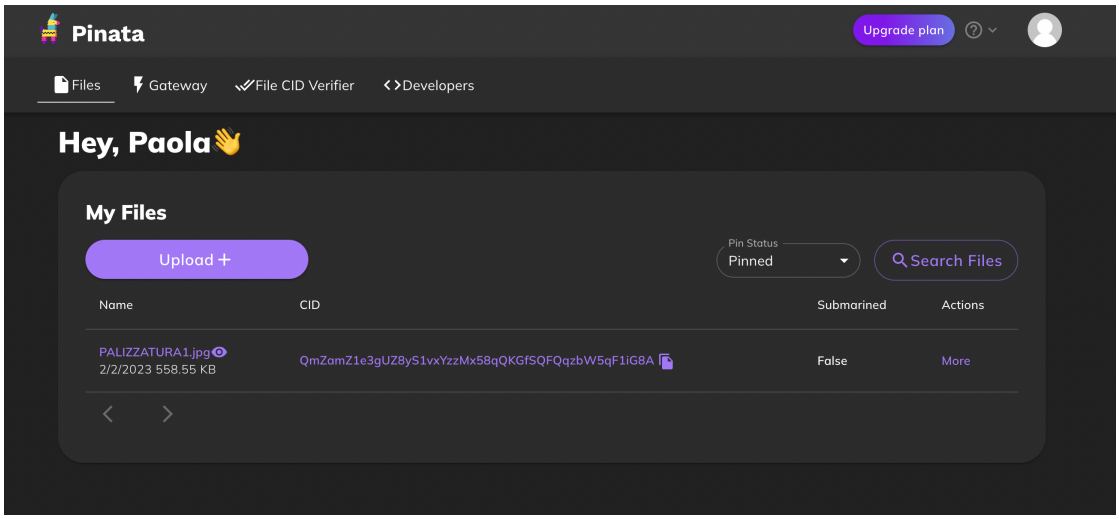

Figura 4.3: Pinata Dashboard.

[5]Nelle telecomunicazioni, una rete di distribuzione dei contenuti (Content Delivery Network) è una rete geograficamente distribuita di server proxy e relativi data centers. L'obiettivo è fornire alta disponibilità e prestazioni distribuendo il servizio in modo spaziale rispetto agli utenti finali.

Una volta all'interno della dashboard, si può scegliere di caricare file, cartelle o CID (identificatori di contenuto) (Figura 4.3).

In generale, Pinata offre prestazioni migliori rispetto all'utilizzo diretto di IPFS, consentendo un recupero più rapido dei dati archiviati. Il CID è una stringa univoca di lettere e numeri che rappresenta un elemento di dati. Non è possibile modificare un CID su una rete IPFS senza modificare il contenuto associato al CID. Per il caricamento su Pinata è sufficiente cliccare su Upload e selezionare il file da caricare. In seguito, la Dashboard sarà aggiornata con l'ultimo oggetto aggiunto affiancato dal CID appena generato, come mostrato in Figura 4.3.

4.6 Metamask

MetaMask [46] è un portafoglio virtuale che consente agli utenti di accedere in modo sicuro all'ecosistema Web3 di applicazioni decentralizzate (dApps). Il wallet è disponibile come applicazione mobile e come estensione del browser su Google Chrome, Firefox, Opera e Brave. Le connessioni possono avvenire su Ethereum o su diverse reti di prova (Figura 4.4). Oggi MetaMask è compatibile con qualsiasi blockchain che esponga un'API JSON RPC compatibile con Ethereum, comprese le blockchain personalizzate e private.

Una volta installato, consente agli utenti di immagazzinare Ether, permettendogli di effettuare transazioni con qualsiasi indirizzo Ethereum. Permette, inoltre, di visualizzare saldi e transazioni, inviare e ricevere fondi, accedere ai portafogli. Grazie alla sua interfaccia intuitiva (Figura 4.5), Metamask è diventato uno strumento indispensabile per gli utenti di criptovaluta, fornendo un accesso rapido e semplice alle applicazioni decentralizzate. Poiché aggiunge funzionalità al normale contesto del browser, MetaMask richiede il permesso di leggere e scrivere su qualsiasi pagina web.

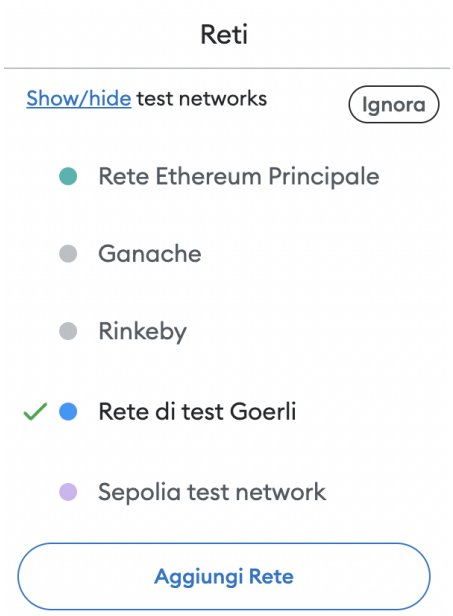

Figura 4.4: Configurazioni reti Metamask.

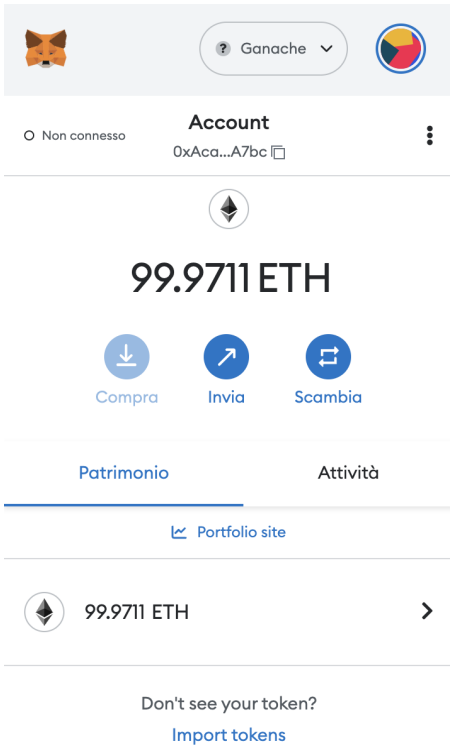

Figura 4.5: Account Metamask.

Per quanto concerne il funzionamento, Metamask inietta un oggetto web3 all'interno del contesto JavaScript dei siti web in esecuzione, consentendo alle dApp un'interfaccia immediata e di interagire direttamente con la blockchain. Inoltre, MetaMask consente anche la gestione degli account agendo come metodo di verifica prima dell'esecuzione di qualsiasi transazione sulla blockchain.

All'utente viene mostrata un'interfaccia per rivedere la transazione, approvarla o rifiutarla prima che possa raggiungere la blockchain di destinazione. Può essere utilizzato per connettersi a una blockchain in esecuzione locale, come Ganache, ed è necessario per accedere in visualizzazione ai dati salvati sulla blockchain oltre che per interagire con la blockchain stessa. Infatti, se i dati richiedono l'interazione con gli smart contract, è necessario l'utilizzo di un portafoglio come MetaMask per eseguire transazioni sulla blockchain.

4.7 Goerli

Goerli, anche noto come Görli, è una rete di test pubblica e distribuita, basata su Ethereum [47]. É stata lanciata nel 2019 e, oggi, rappresenta una delle più importanti testnet. É uno strumento fondamentale per testare aplicazioni decentralizzate e smart contract prima di implementarli sulla rete principale di Ethereum. Nel presente progetto di tesi è stata impiegata perchè ben integrata con molti strumenti di sviluppo come Remix, Truffle e Ganache ed è anche supportata da numerosi provider di servizi blockchain, tra cui Infura.

4.7.1 Ganache e Goerli a confronto

Si è deciso di testare la dAapp anche su Goerli oltre che su Ganache perchè, se quest'ultima è un ambiente di sviluppo blockchain locale che simula una blockchain Ethereum su un computer locale, Goerli è una vera e propria rete di test sulla

blockchain Ethereum che consente di testare le applicazioni in un ambiente più simile alla rete principale.

Inoltre, utilizzando una vera rete di test come Goerli, è possibile testare l' applicazione in un ambiente più realistico, con una maggiore sicurezza e un maggior livello di affidabilità rispetto a un ambiente di sviluppo locale come Ganache.

Tuttavia, utilizzare Ganache in una prima fase di test è stato fondamentale perchè risulta un'opzione più leggera e veloce. La testnet Goerli, invece, risulta più lenta in quanto la velocità di elaborazione delle transazioni è influenzata dalla congestione della rete, dall'affidabilità dei nodi, dall'uso di risorse di calcolo da parte degli utenti e dal fatto che sulla rete di test Goerli le transazioni devono essere propagate e validate da nodi della rete che si trovano in tutto il mondo.

In una seconda fase, ricorrere a Goerli è stato fondamentale per testare l'applicazione blockchain che richiede l'interazione tra più dispositivi. Infatti, con Goerli i dati generati durante i test sono accessibili ad altri dispositivi che si connettono alla stessa rete di test. É stato così possibile verificare il corretto funzionamento dell'applicazione in un ambiente distribuito e realistico. Questo è possibile grazie alla natura decentralizzata della blockchain Ethereum, che consente ai nodi della rete di replicare e sincronizzare i dati in modo autonomo.

4.7.2 Infura

Il modo più semplice per raggiungere una testnet è utilizzare un provider, e, in particolare, per il presente progetto, è stato utilizzato Infura [48]. Infura è un servizio che permette l'accesso gratuito a un nodo della rete Ethereum, richiede solo la registrazione per ottenere una chiave e l'RPC URL relativo alla rete a cui ci si vuole connettere. In alternativa ad Infura, è possibile avviare il proprio nodo con tool specifici(come Geth) ma questa opzione è stata scartata in quanto richiede un un intenso uso di risorse (Geth richiede notevoli risorse di calcolo per funzionare)

e, in generale, risulta meno efficiente per l'obiettivo del progetto di tesi, ovvero la creazione di un'applicazione.

Capitolo 5

Prototipo realizzato

In seguito a un'analisi approfondita dei requisiti volta a identificare con precisione i requisiti funzionali e non funzionali e a uno studio dettagliato delle tecnologie necessarie per soddisfarli, è stato possibile procedere con l'implementazione della dApp.

In questo capitolo vengono mostrati i dettagli implementativi e le funzionalità più importanti del prototipo pensato per l'azienda vitivinicola. Come accennato nei capitoli precedenti, per simulare la rete di Ethereum nella fase di testing è stata utilizzata prima Ganache per poi passare a Goerli.

5.1 Architettura del sistema

Il primo step è l'individuazione dell'architettura del sistema. Le tecnologie impiegate sono ampiamente descritte nel capitolo 4 e esplicitate nella Figura 5.1. Riassumendo, l'architettura progettuale [49] si basa sulla blockchain di Ethereum, utilizza Pinata come strumento di pinning di IPFS e la Web Application.

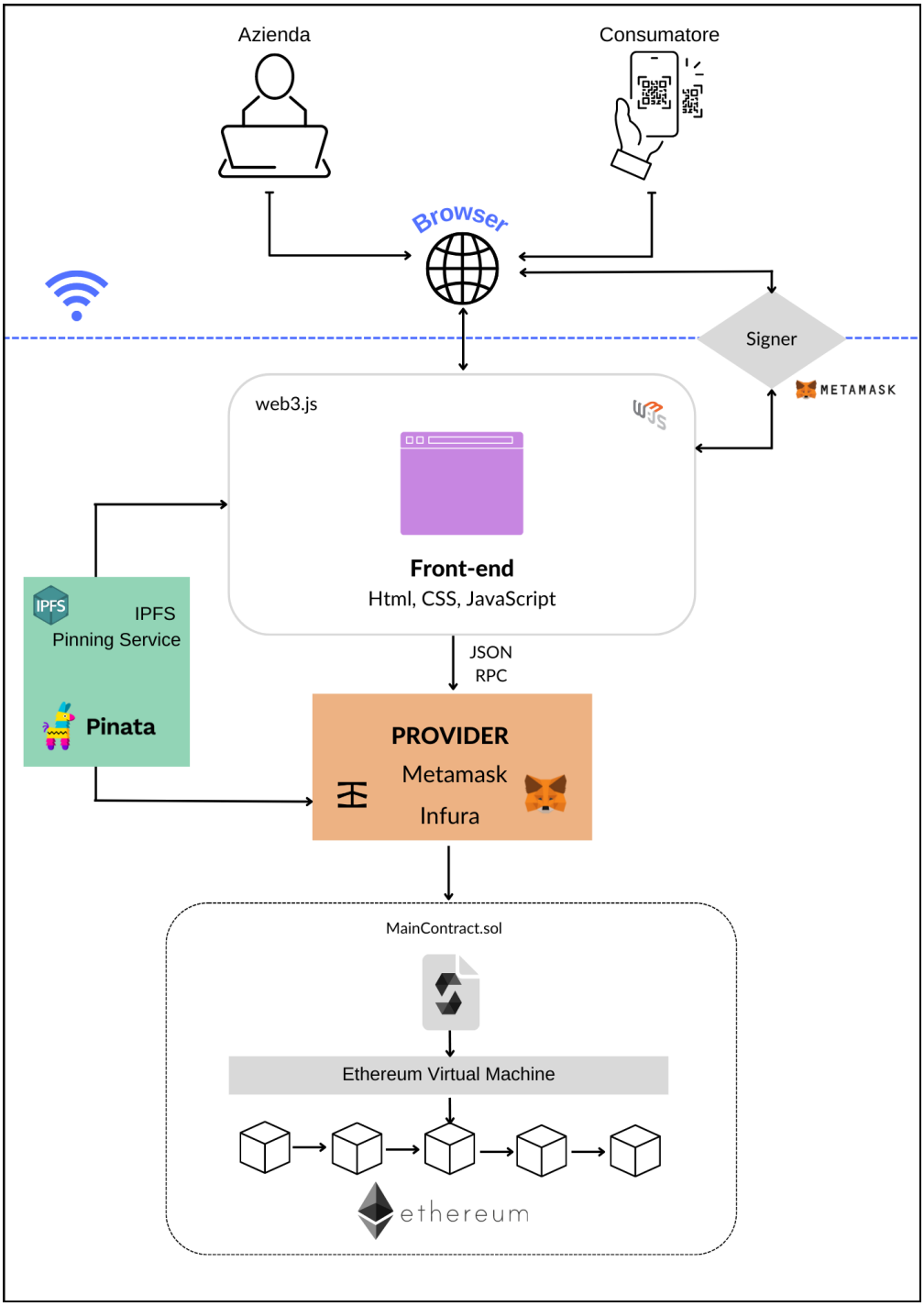

Figura 5.1: Architettura del sistema.

5.2 Back-end

5.2.1 Sviluppo dello Smart Contract

Il punto di partenza della fase implementativa è rappresentato dalla stesura dello *smart contract*, realizzato utilizzando il linguaggio Solidity che, una volta distribuito sulla rete, risulta accessibile a livello globale con elevata integrità, disponibilità, resilienza e trasparenza.

Ogni dato che viene salvato richiede spazio sulla blockchain e quindi richiede il pagamento di gas per archiviarlo. Infatti, in un primo momento è stato valutato, in accordo con la start-up Sblockchain Project, tra i vari dati a disposizione, quali salvare e quali, invece, omettere (Tabella 5.1) .

La realizzazione dello smart contract, dunque, è stata progettata in modo da minimizzare il consumo di gas senza compromettere la funzionalità o la sicurezza dello stesso. Per garantire ciò, sono state simulate diverse soluzioni possibili tra cui:

- realizzare un unico smart contract;
- realizzare uno smart contract principale che ne estende altri tre, ereditandone tutte le funzionalità.

In un primo momento si è optato per la seconda soluzione per poter sviluppare l'applicazione in modo modulare e scalabile. Con l'ereditarietà (relazione modellata attraverso la relationship "is"), uno smart contract può acquisire funzionalità e proprietà da uno o più smart contract, consentendo di creare una gerarchia di contratti che facilita l'implementazione di funzionalità complesse. Ma, se da una parte, l'uso di contratti ereditati può semplificare lo sviluppo degli smart contract, dall'altra può aumentare il consumo di gas in quanto aumenta il numero di operazioni di scrittura e di lettura su blockchain.

Quindi, se utilizzare contratti ereditari semplifica la progettazione del contratto permettendo di suddividere le funzionalità in moduli separati e mantenere il codice più organizzato, risulta sconveniente quando è necessario salvare una grande quantità di dati per il maggiore consumo di gas. D'altra parte, l'utilizzo di un unico smart contract può ridurre il consumo di gas, poiché tutte le funzioni sono raggruppate in un unico contratto e le transazioni vengono eseguite in un unico blocco. In conclusione si è optato per la prima soluzione: è stato realizzato un unico smart contract.

Il contratto realizzato è relativo alla creazione di un prodotto rappresentato dal lotto di vino della stessa annata. Il contratto, quando invocato, permetterà di salvare le informazioni relative all'oggetto su Ethereum.

Lo smart contract prevede: quattro struct, un evento, un costruttore, un modificatore, delle funzioni per il salvataggio e altre funzioni per interrogare la blockchain e ottenerne i dati. Una *struct* rappresenta una struttura dati composta da uno o più campi, in cui sono state inserite tutte le informazioni necessarie per la definizione del prodotto. La struct principale è quella del prodotto che prevede:

- un intero che rappresenta l'id del prodotto;
- un intero per salvare il momento di aggiunta del prodotto su blockchain;
- un oggetto di tipo *Dettagli*;
- un oggetto di tipo *Contesto*;
- un oggetto di tipo *Processo*;
- una stringa che rappresenta la certificazione etica e ambientale;
- una stringa che rappresenta un'immagine.

```
48    struct Prodotto {
49      uint id;
50      uint time;
51      string immagine;
52      string documento;
53      Dettagli d;
54      Contesto c;
55      Processo p;
56    }
```

Figura 5.2: Struct *Prodotto*.

Il momento in cui avviene salvataggio dei dati relativi al prodotto su blockchain, viene salvato per mezzo di *block.timestamp*[1], variabile predefinita in Solidity che restituisce il timestamp del blocco corrente, in secondi trascorsi dall'*Epoch Unix* (01/01/1970) fino al momento in cui il blocco è stato creato. Gli oggetti di tipo Dettagli, Contesto e Processo sono a loro volta definiti come struct nello stesso smart contract: Solidity permette di richiamare altre struct all'interno di una struct e, per farlo, è sufficiente definirla all'interno della struttura principale. L'utilizzo di struct nidificate consente di organizzare meglio i dati all'interno dello smart contract, migliorando la leggibilità del codice e semplificando la gestione dei dati. Le struct Dettagli, Contesto e Processo sono state definite principalmente per mezzo di stringhe e sono costituite rispettivamente da otto, cinque e quattro campi.

L'azienda ha fornito una serie di informazioni ma, dato l'elevato numero di dati, sono state fatte delle scelte stringenti in termini di spazio di memoria, in quanto, con tutti i dati richiesti, venivano superati i limiti di gas previsti. Per questo motivo, si è optato per aggregare alcuni dati. Ad esempio, l'azienda richiedeva due campi *grado alcolico* e *caratteristicheChimiche* e si è scelto di aggregarli sotto un'unica voce, stessa cosa accaduta per le variabili *abbinamentiGastronomici* e *consigliDUso*.

[1]https://docs.soliditylang.org/en/latest/units-and-global-variables.html#block-and-transaction-properties

In particolare, dovendo gestire una grande quantità di dati nella funzione *createProduct*, è stato fondamentale suddividere il processo in sotto-funzioni più specifiche e facilmente controllabili in modo da ridurne la complessità, il rischio di errori o di perdita di dati, semplificando l'implementazione. Il problema di creazione del prodotto è stato pertanto ridotto a tre funzioni: *addDettagliProdotto*, *addDettagliContesto*, *addDettagliProcesso* che permettono il salvataggio, rispettivamente, degli oggetti Dettagli, Contesto e Processo.

Il costruttore è una funzione dichiarata con la parola chiave *constructor*, utilizzata per inizializzare le variabili di stato di un contratto. Ogni contratto può avere un solo costruttore, viene eseguito una sola volta ossia quando viene creato e distribuito sulla rete. Nel presente lavoro, il costruttore è stato utilizzato per impostare l'inidirizzo dell'account che ha creato il contratto come proprietario, per mezzo della variabile predefinita *msg.sender*.

Per verificare che solo il proprietario possa chiamare alcune funzioni del contratto, impedendo ad altri account di accedere a queste funzionalità, è stato aggiunto un modificatore, *onlyOwner()*. Nello smart contract, le funzioni che può richiamare solo l'owner sono quelle di aggiunta dei dati quindi: *addDettagliProdotto* (Figura 5.3), *addDettagliContesto*, *addDettagliProcesso* e *createProduct*. Le funzioni per la visualizzazione, invece, sono accessibili a tutti.

```
67    function addDettagliProdotto(string memory _nome,
68      string memory _denominazione, string memory _certificato,
69      string memory _composizione,
70      string memory _caratteristicheChimiche,
71      string memory _caratteristicheOrganolettiche, string memory _consigliduso)
72
73     public onlyOwner{
74      dettagli = Dettagli(_nome, _denominazione, _certificato, _composizione, _caratteristicheChimiche,
75        _caratteristicheOrganolettiche, _consigliduso
76        );
77    }
```

Figura 5.3: Funzione *addDettagliProdotto*.

Per permettere all'utente di memorizzare più prodotti, è stato utilizzato un *mapping*. La mappatura in Solidity permette di associare una chiave a un valore.

Nel mapping la chiave è rappresentata da un valore di tipo *uint* (che rappresenta l'id del prodotto, incrementato ogni volta che viene richiamata la funzione creatProduct per mezzo di un contatore(*productCount*)) e il valore è prodotto stesso. Il contatore è una variabile che si aggiorna sempre di una quantità costante. Il mapping ha permesso di accedere facilmente nella fase successiva al prodotto conoscendone solo l'id. Si è optato per un mapping e non per un array perché l'obiettivo non era poter iterare sul gruppo di dati ma voler recuperare i valori in base a una chiave nota. Sono state infine definite delle *get-functions* per restituire i dati nella pagina di visualizzazione. Di seguito è riportata la funzione *getContesto* come esempio.

```
118   function getContesto(uint id) public view returns (
119     string memory formaDiAllevamento,
120     string memory etaVigneti,
121     string memory estensioneVigneti,
122     string memory terreno,
123     string memory resaEttaro) {
124
125     Prodotto memory product = products[id];
126
127     return (product.c.formaDiAllevamento, product.c.etaVigneti,
128       product.c.estensioneVigneti, product.c.terreno, product.c.resaEttaro);
129
130   }
```

Figura 5.4: Funzione *getContesto*.

I valori salvati e le funzioni sono state definite come *pubbliche* così da renderle disponibili anche all'esterno del contratto: altri contratti o utenti possono accedere alle variabili o funzioni pubbliche tramite chiamate di funzione. Le get-functions prendono in input l'id del prodotto, recuperato tramite il mapping, e resituiscono i dati di interesse.

Test su GANACHE	**Dati**	**Tipo**	**Tentativo n.1**	**Tentativo n.2**	**Tentativo n.3**
Dettagli	nome	string	x	x	
	denominazione	string	x	x	x
	certificato	string	x	x	
	composizione	string	x	x	
	gradoAlcolico	uint	x		
	caratteristicheChimiche	string	x	x	x
	caratteristicheOrganolettiche	string	x	x	x
	consigliUso	string	x	x	x
	abbinamentiGastronomici	string	x		
Contesto	formaDiAllevamento;	string	x	x	x
	etaVigneti	string	x	x	x
	estensioneVigneti	string	x	x	x
	terreno	string	x	x	x
	resaEttaro	string	x	x	x
Processo	vinificazioneProduzione	string	x	x	x
	numbottiglieProdotte	string	x	x	x
	immagineproduzione	string	x	x	x
	descrizione	string	x	x	
Prodotto	id	uint	x	x	x
	time	uint	x	x	
	immagine	string	x	x	x
	documento	string	x	x	x
Totale dati considerati			22	20	15
Gas Utilizzato			6721975	5064645	3749575
Costo in ETH				0.1012929	0.07996858
Costo in Euro				159,6	126

Tabella 5.1: Valutazione del consumo di gas dello smart contract in base al numero di dati considerati.

5.2.2 Interazione con il Front-End

L'interazione tra back-end e front-end è possibile grazie alla libreria *Web3.js* (paragrafo 4.4). Il codice associato è il seguente:

```
10
11    initWeb3: async function() {
12      if (window.ethereum) {
13        App.web3Provider = window.ethereum;
14        try {
15          // Richiede l'accesso all'account
16          await window.ethereum.request({ method: 'eth_requestAccounts' });
17          web3 = new Web3(window.ethereum);
18          console.log("stato connessione ethereum " + window.ethereum.isConnected());
19          web3.eth.getCoinbase(function(err, account) {
20            if (err === null) {
21              App.account = account;
22              //Mostra l'indirizzo dell'account
23              console.log("Your Account: " + account);
24            }
25          });
26          return App.initContract();
27
28        } catch (error) {
29          console.error("User denied account access")
30        }
31      }
32      else if (typeof web3 !== 'undefined') {
33        // se un'istanza web3 è già fornita da Meta Mask.
34        App.web3Provider = web3.currentProvider;
35        web3 = new Web3(web3.currentProvider);
36      } else {
37        // Specifica l'istanza predefinita se non è stata fornita un'istanza web3
38        App.web3Provider = new Web3.providers.HttpProvider('http://localhost:7545');
39        web3 = new Web3(App.web3Provider);
40      }
41      return App.initContract();
42    },
43
```

Figura 5.5: Funzione *initWeb3*.

Nella Figura 5.5, la funzione *initWeb3* viene chiamata per inizializzare la connessione a Web3. Questa funzione controlla se il browser ha a disposizione l'oggetto provider Ethereum (*if(window.ethereum)*) e, se l'esito del controllo è positivo, significa che è presente un portafoglio in grado di comunicare con la rete: viene creata una nuova istanza Web3 utilizzando il provider corrente.

Se, invece, l'oggetto provider Ethereum non è disponibile, ci sono due possibilità:

1. web3 è già stato definito;

2. web3 non è stato definito e *App.web3Provider* viene impostato su un oggetto

HttpProvider legato a una blockchain di sviluppo locale in esecuzione su *http://localhost:7545*.

La funzione restituisce *App.initContract()*, funzione che inizializza il contratto intelligente utilizzato dalla dApp per interagire con la blockchain.

```
49    initContract: function() {
50
51      $.getJSON("MainContract.json", function(mainContract) {
52        App.contracts.MainContract = TruffleContract(mainContract);
53
54        // Connette il provider per interagire con il contratto
55        App.contracts.MainContract.setProvider(App.web3Provider);
56
57        return App.render();
58      });
59    },
```

Figura 5.6: Funzione *initContract*.

5.3 Front-end

Se la logica back-end è stata realizzata per mezzo degli smart contract, l'architettura del frontend, nel caso delle DApp, è incentrata sulla comunicazione con gli smart contract in quanto fornisce un'interfaccia utente per interagire con essi e recuperare i dati salvati. Il frontend di una DApp è costruito in modo molto simile a un'applicazione Web tradizionale (utilizzando un mix di HTML, CSS e JavaScript), ma permette l'interazione diretta con la blockchain. Per lo sviluppo dell'interfaccia che compone la Dapp, si è optato per lo sviluppo in JavaScript nella sezione sezione Frontend e di HTML e CSS per organizzare e strutturare il contenuto dell'interfaccia utente. Sono state inoltre utilizzate le librerie Bootstrap per rendere l'interfaccia un po' più intuitiva e minimalista [50]. Il front-end della dApp presenta due interfacce distinte:

- la prima è quella di input, pensata per permettere all'utente di inserire autonomamente i dati necessari per salvare le informazioni su blockchain;

- la seconda,invece, ha un duplice compito, è la pagina a cui viene reindirizzato l'utente che ha inserito i dati ed è la pagina di visualizzazione per il consumatore finale.

La gestione delle due pagine HTML associate allo stesso file JavaScript avviene nella funzione render(). Nella funzione, infatti, viene definito cosa deve essere visualizzato su ciascuna interfaccia.

5.3.1 Interfaccia per l'aggiunta del prodotto

Per la creazione del prodotto e il salvataggio dei relativi dati, è stata realizzata un'interfaccia che permetta di inserire e salvare informazioni su blockchain pubblica. Per visualizzare il contenuto della pagina web è necessario sbloccare il portafoglio virtuale e connettersi alla rete. Solo dopo essersi connessi, può essere mostrato il contenuto della pagina. Il contenuto della pagina di definizione del prodotto è resa disponibile solo se l'account che vuole accedere alla pagina corrisponde a quello definito. L'indirizzo dell'account con cui si stanno eseguendo le transazioni è visibile in fondo alla pagina. Il controllo sull'account è stato gestito nel file App.js come segue:

```
100   controlloAccount: async function(){
101
102     const accounts = await ethereum.request({ method: 'eth_accounts' });
103     const accountCorrente = accounts[0];
104
105     // Indirizzo dell'account che può accedere alla pagina di inserimento dati
106     const accountRichiesto = '0xaca58df159b76b120f7053a1ca410eb0f680a7bc';
107
108     if (accountCorrente !== accountRichiesto) {
109       return false;
110     }
111     else
112       return true;
113   },
```

Figura 5.7: Funzione *controlloAccount*.

Nel frammento di codice viene richiamata la funzione *ethereum.request* per richiedere l'elenco degli account Ethereum disponibili su Metamask. Alla funzione

viene assehnato il metodo *method: 'eth_accounts'*, definito nell'API Web3 Ethereum che restituisce un array di stringhe rappresentanti gli indirizzi degli account sbloccati dell'utente. L'array di stringhe viene assegnato alla costante *accounts* per poterne estrarre il valore dell'account corrente (corrispondente al primo valore della costante). Se l'account corrente non corrisponde a quello richiesto, la funzione restituisce *false* e viene mostrato sulla schermata un messaggio di errore. Invece, se l'esito della funzione è positivo, viene mostrato il contenuto della pagina. L'estensione di Metamask permette però di poter cambiare account continuamente, è stato quindi necessario definire un'ulteriore funzione che gestisca questa eventualità. Questo aspetto è stato gestito nel codice Javascript come segue:

window.ethereum.on('accountsChanged', App.handleAccountsChanged);

Quando si verifica l'evento *accountsChanged* [2], viene attivata la funzione *handleAccountsChanged* che, al variare dell'account, aggiorna la pagina corrente e reindirizza alla funzione *"controlloAccount"*. L'evento è messo a disposizione dal provider MetaMask e viene emesso ogni volta che l'indirizzo dell'account esposto dell'utente cambia. In questo modo è possibile aggiornare lo stato dell'applicazione in base all'account sbloccato.

Confermata la correttezza dell'account che tenta di accedere, all'utente verrà mostrata l'interfaccia riportata nella Figura 5.8.

La piattaforma è stata pensata per l'azienda cliente, quindi è stata appositamente concepita per offrire un'esperienza d'uso agevole e intuitiva, fornendo una soluzione chiara e di facile utilizzo. Con questo fine, l'inserimento dei dati è stato suddiviso in quattro step distinti:

1. Step 1: per inserire i dettagli del prodotto;
2. Step 2: per inserire i dettagli sul contesto;

[2]https://docs.metamask.io/guide/ethereum-provider.html#using-the-provider

3. Step 3: per aggiungere le informazioni sul processo;

4. Step 4: per aggiungere ulteriori file.

Gli step sono stati definiti nel file html per l'inserimento dei dati (*index.html*). Al click sullo step è associata la funzione *toggleForm(id)* che mostra i campi di input da compilare solo dopo il click sul form. Nel form sono disponibili due pulsanti, uno di tipo *submit* (Invia) e l'altro di tipo *reset* (Cancella).

Figura 5.8: Schermata dApp: definizione prodotto.

Di seguito è riportata l'impostazione di due step come esempio.

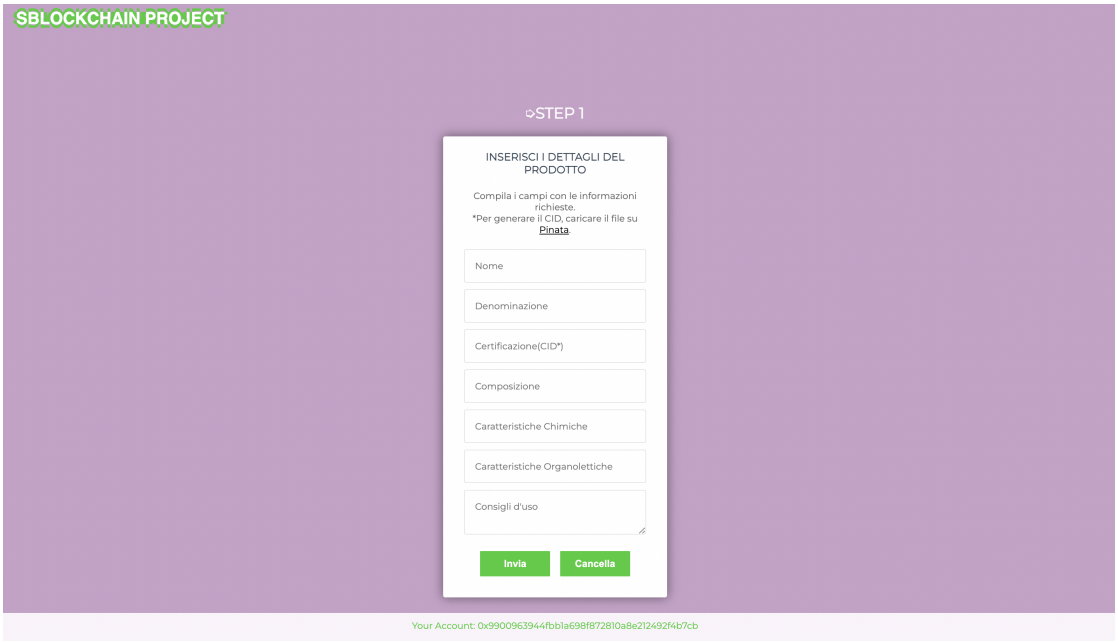

Figura 5.9: Schermata dApp: primo step.

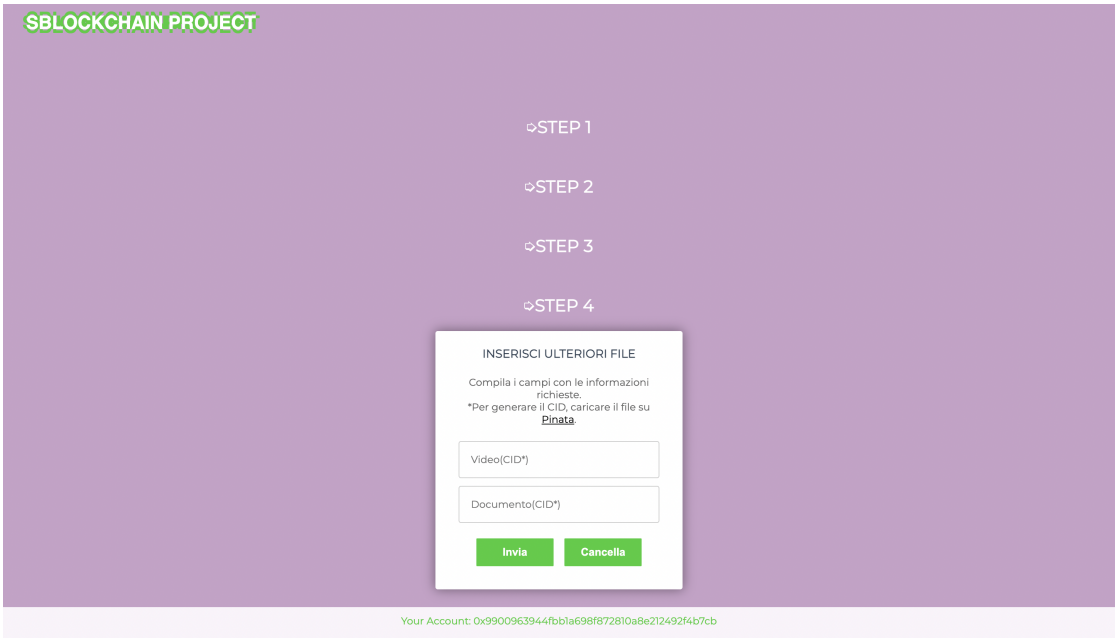

Figura 5.10: Schermata dApp: quarto step.

In base allo step selezionato, al click sul pulsante di tipo submit sono associate

le funzioni Javascript: *App.DettagliProdotto()*, *App.Contesto()*, *App.Processo()* e *App.creaProdotto()*. Se uno dei campi risulta vuoto, viene restituito un messaggio di errore. Le funzioni *App.DettagliProdotto()*, *App.Contesto()*, *App.Processo()* e *App.creaProdotto()*, sono strutturate in modo simile e di seguito viene riporatata la funzione *Contesto* come esempio:

```
contesto: function(){

  var statoInstance;

  var formaDiAllevamento = $('#textbox5').val();
  var etaVigneti = $('#textbox6').val();
  var estensioneVigneti = $('#textbox7').val();
  var terreno = $('#textbox8').val();
  var resaEttaro =  $('#textbox9').val();

   if (!formaDiAllevamento || !etaVigneti || !estensioneVigneti || !terreno || !resaEttaro) {
    alert("Per favore, compila tutti i campi.");
     return false;
   }
   else {

    App.contracts.MainContract.deployed().then(function(instance) {

    statoInstance = instance;

    return statoInstance.aggiungiContesto(formaDiAllevamento, etaVigneti, estensioneVigneti,
      terreno, resaEttaro,{ from: App.account });
    }).then(function(result) {
      console.log(result);
      console.log("Dettagli Contesto aggiunti correttamente");

      const form = document.getElementById("nascondi2");
      const successMessage = document.getElementById("messaggio2");
      form.style.display = "none";
      successMessage.style.display = "block";

    }).catch(function(error) {
       console.warn(error);
    });
  }
},
```

Figura 5.11: Funzione *Contesto*.

Il codice in Figura 5.11 esegue una transazione sulla blockchain per mezzo dello smart contract *MainContract*. Il metodo “deployed()” su MainContract richiama un’istanza del contratto dal nodo a cui l’applicazione è connessa. Successivamente, sull’istanza del contratto viene richiamata la funzione *aggiungiContesto* per salvare i valori inseriti dall’utente nelle caselle di input del file HTML. Questa transazione viene inviata dalla posizione dell’account corrente dell’applicazione, specificato come parametro nella chiamata “from: App.account”. Infine, la promessa viene risolta quando la transazione viene confermata dal nodo Ethereum e restituisce il risultato della transazione.

Se tutti i valori sono stati inseriti correttamente, Metamask si attiverà automaticamente e chiederà di confermare la transazione mostrandone tutti i dettagli: il mittente, il destinatario, la commissione di rete e l'importo. All'utente è data la possibilità di confermare o annullare la transazione come mostrato in Figura 5.12.

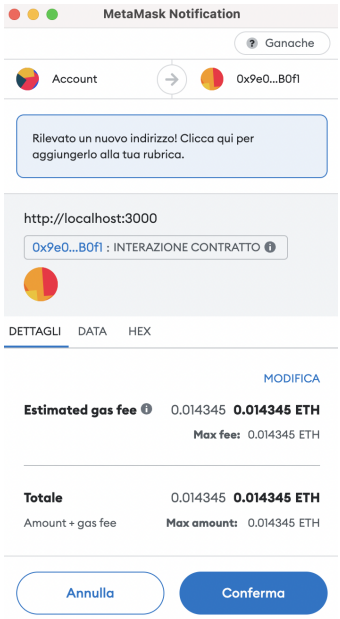

Figura 5.12: Finestra di dialogo Metamask per approvare la transazione su rete locale Ganache.

Confermata la prima transazione tramite Metamask, bisognerà fare lo stesso con le transazioni successive fino all'ultimo step che consentirà la creazione del prodotto: la funzione *createProduct* verrà richiamata solo con la conferma dell'ultima transazione. Ad ogni step è richiesto l'inserimento di un numero variabile di valori di input. Se uno dei campi risulta vuoto, viene restituito un messaggio di errore che invita l'utente a completare tutti i campi. Negli step 1, 3 e 4 viene inoltre richiesto l'inserimento di un CID da ottenere tramite la piattaforma Pinata (come spiegato nel paragrafo 4.5.1): nello smart contract verrà salvato esclusivamente il CID del file come stringa. Ad ogni transazione confermata e quindi ad uno step concluso con successo, è associato un messaggio di successo.

Dopo aver confermato la transazione, Metamask la invia alla rete Ethereum per la conferma. La transazione viene quindi visualizzata nella cronologia delle transazioni di Metamask, insieme ad uno stato "Pending" che indica che la transazione è in attesa di essere confermata dalla rete.
La conferma della transazione avviene attraverso il processo di mining. I miner della rete Ethereum utilizzano la potenza di calcolo per risolvere problemi crittografici e verificare la validità delle transazioni. Quando la transazione viene confermata, lo stato della transazione cambia da "Pending" a "Confirmed" nella cronologia delle transazioni di Metamask. Una volta che la transazione è confermata, gli ether associati vengono inviati al destinatario.

5.3.2 Interfaccia per la visualizzazione del prodotto

Oltre alla pagina di definizione del prodotto, è presente un'altra pagina: *Visualizza*. A differenza della pagina *index.html* pensata esclusivamente per il produttore, la pagina di visualizzazione è resa disponibile sia per il produttore che per il consumatore finale.

Il produttore viene indirizzato alla pagina *visualizza.html* subito dopo aver aggiunto il prodotto con successo, quindi dopo aver completato l'ultimo step. Il consumatore finale accederà invece alla pagina di visualizzazione tramite un QR-code letto dall'app di Metamask (disponibile sia su GooglePlay che su AppStore).

Dopo aver creato con successo il prodotto, l'utente accede alla pagina di visualizzazione per mezzo del metodo *window.location.href = "visualizza.html";* richiamato appena la funzione di creazione del prodotto si conclude con successo. Nell'URL della pagina a cui si viene reindirizzati, viene aggiunto un id aggiornato ad ogni aggiunta del prodotto. Con l'aggiunta del prodotto, ad esempio del prodotto 1, l'URL diventerà: *http://localhost:3000/visualizza.html?id=1*.

Il cuore centrale dell'interfaccia è rappresentato dalla funzione *display* in cui:

- viene estratto l'id dall'Url, reso possibile dalla funzione *split*;

- viene richiamata un'istanza del contratto MainContract e letto il productCount;

- si verifica che l'id estratto sia minore o uguale al productCount, se ciò non accade verrà restituito un messaggio di errore in quanto il prodotto con l'ID inserito non esiste o non è stato ancora definito e si viene reindirizzati alla pagina di definizione del prodotto;

- se la verifica del punto precedente si è conclusa con successo, vengono richiamate le get-functions del contratto e, per mezzo di cicli for, i campi della pagina di visualizzazione vengono riempiti;

- vengono costruite le stringe per concatenare i CID salvati negli smart contract al resto dell'url per accedere ai file tramite Pinata;

- viene trasformato il timestamp, memorizzato dallo smart contract come uint in secondi, in un oggetto di tipo *Date*.

L'interfaccia prevede un elenco dei dati salvati con le relative immagini e documenti. La visualizzazione è resa possibile grazie alla funzione *display* definita in JavaScript.

Di seguito tre schermate della dAPP in visualizzazione:

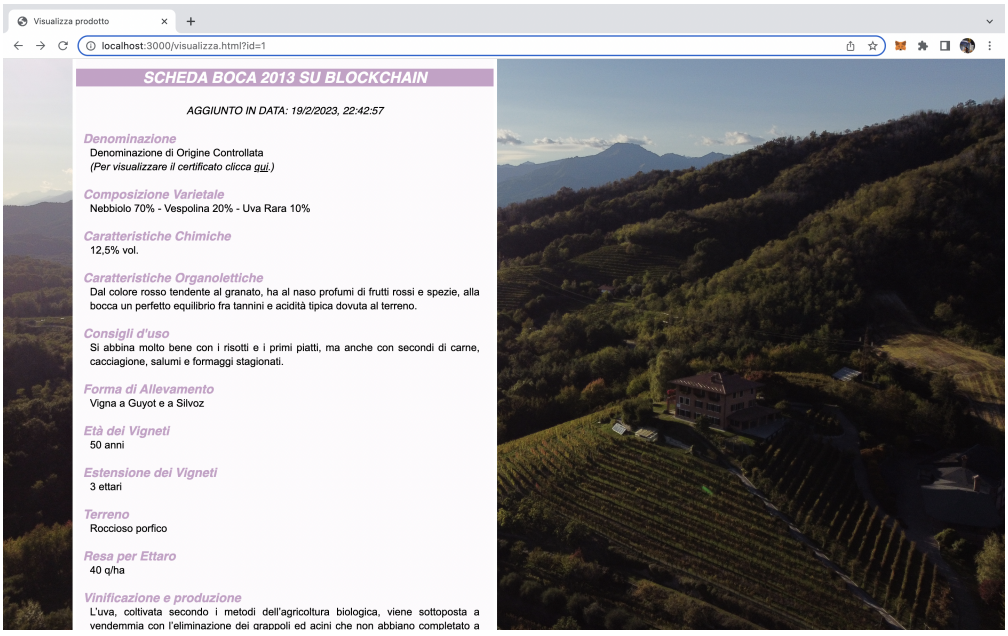

Figura 5.13: Schermata DApp: interfaccia di visualizzazione, prima parte.

Figura 5.14: Schermata dApp: interfaccia di visualizzazione, seconda parte.

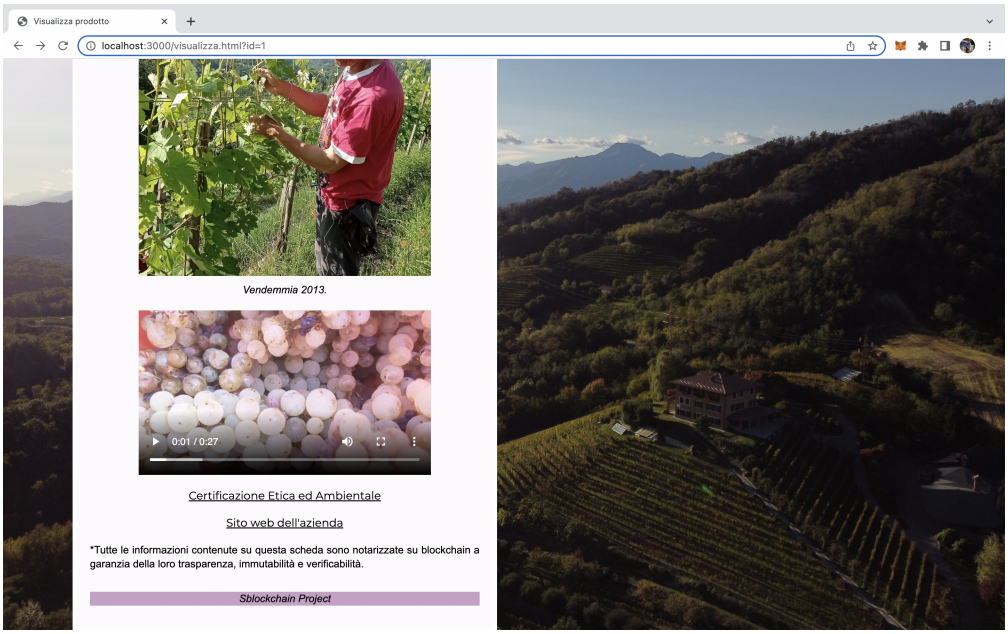

Figura 5.15: Schermata dApp: interfaccia di visualizzazione, terza parte.

All'aggiunta di un nuovo prodotto, l'ID della pagina di visualizzazione verrà aggiornato e sarà possibile passare da un prodotto a un altro modificando manualmente il valore nell'URL. Ad esempio, aggiungendo un secondo prodotto nel sistema, si otterrà:

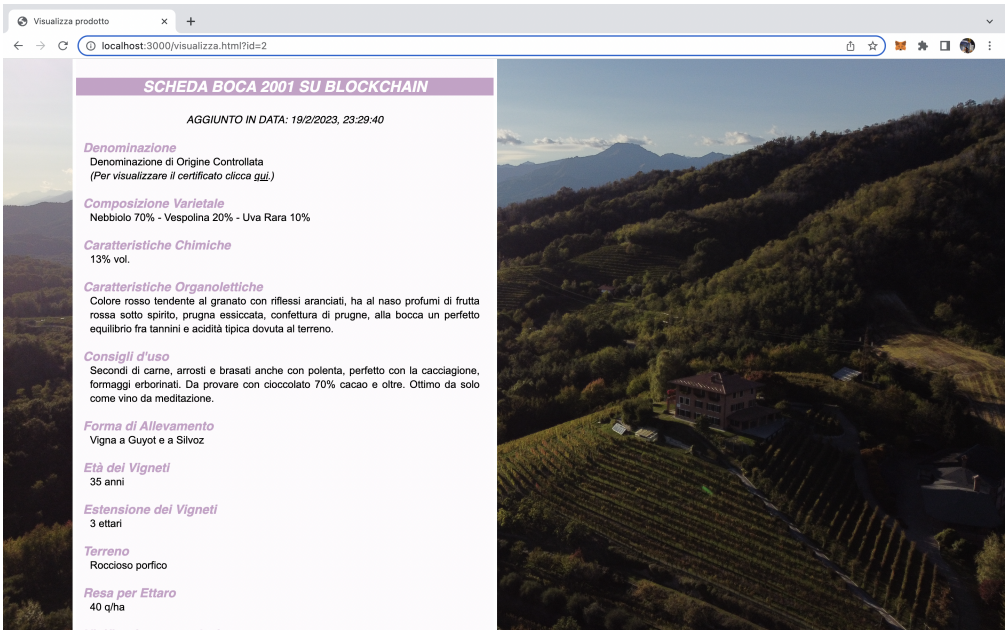

Figura 5.16: Schermata dApp: aggiornamento URL.

5.4 Dimostrazione su altri dispositivi

Dopo aver sviluppato la dApp e averne accertato il funzionamento in locale, è stato necessario apportare delle modifiche per poter dimostrare il suo completo funzionamento. Questo processo è importante per dimostrare che la dApp funziona correttamente su una varietà di reti e dispositivi e per ottenere il feedback dell'azienda.

5.4.1 Configurazione Rete di Test Goerli

Per poter mostrare il funzionamento della DApp su più dispositivi, è stato necessario spostarsi da Ganache, un ambiente di sviluppo locale, a Goerli, una rete di test pubblica. Dopo aver testato la DApp e averne accertato il corretto funzionamento sul nodo locale di Ganache, si è deciso di migrare la stessa sulla blockchain di Goerli per testarne le funzionalità in un ambiente più simile a quello di Ethereum.

Questa migrazione ha richiesto una modifica del file di configurazione, in modo da consentire alla DApp di interagire con la rete di test. Inoltre, prima di distribuire la DApp sulla blockchain di Goerli, è stato necessario valutare il costo del gas richiesto per l'esecuzione delle transazioni sulla blockchain attraverso Etherscan[3]. Etherscan fornisce informazioni sulla blockchain di Ethereum e sui costi di transazione associati. Infatti, il *gas tracker* di Etherscan permette di controllare i prezzi correnti del gas sulla rete. É uno strumento fondamentale per prevedere quanto sarà congestionata la rete e quanto costerà un semplice trasferimento o un'interazione con un contratto intelligente. La corretta valutazione del costo del gas tramite Etherscan prima di distribuire i contratti sulla rete ha permesso un rapporto tempi/costi ragionevole, permettendo di ottimizzare l'efficienza della dApp.

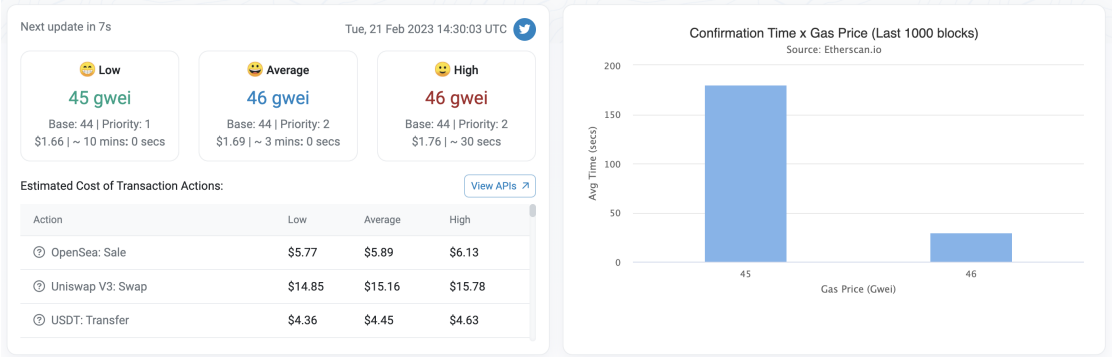

Figura 5.17: Etherscan GasTracker

Configurazione del file *truffle-config.js* per compilare gli smart contract su Goerli con le stime di gas price riportate in Figura 5.17:

[3]https://etherscan.io/

```
module.exports = {

  networks: {

    goerli: {
      provider: () =>
        new HDWalletProvider(
          MNEMONIC,
          `https://goerli.infura.io/v3/20cfa5645aa945cfa412caa014d53db7`
        ),
      network_id: 5,
      from: "0x9900963944FbB1a698F872810a8e212492F4B7cB",
      gas: 15000000,
      gasPrice:45000000000,
      skipDryRun: true
    },
    develop: {
      port: 8545
    },
  },

  compilers: {
     solc: {
        version: "0.8.7",
     }
  }
};
```

Figura 5.18: Configurazione Rete di Test Goerli

Nella Figura 5.19 HDWalletProvider è un provider di Web3.js per Truffle che consente di accedere a una blockchain utilizzando una frase mnemonica (MNEMINIC) e un endpoint di una rete. HDWalletProvider gestisce la creazione delle chiavi private a partire dalla frase mnemonica e fornisce un'istanza Web3.js con cui interagire con la blockchain, il che rende la connessione alla blockchain più sicura. Nella configurazione di Truffle è stato utilizzato Infura come provider di rete per il deployment dello smart contract e l'interazione con la blockchain, al posto di Geth[4] che risulta meno comodo da utilizzare per scopi di testing e sviluppo. In definitiva, la migrazione da Ganache a Goerli ha permesso di verificare la funzionalità della DApp in un ambiente di produzione reale e di garantire un'esperienza utente ottimale.

[4]Go-ethereum (anche noto come Geth) è un client Ethereum costruito in Go. È uno dei client Ethereum originali e più popolari.

5.4.2 Simulazione server

Al fine di mostrare il funzionamento del prototipo su altri dispositivi, in particolare sugli smartphone, necessari per la lettura del QR-code, senza disporre di un servizio di hosting, si è optato per utilizzare l'indirizzo IP del computer come server temporaneo. Questa soluzione è stata scelta come alternativa provvisoria per rendere la pagina accessibile ad altri utenti senza dover necessariamente ospitare il sito web su un server esterno a pagamento. L'utilizzo di un server temporaneo tramite l'indirizzo IP del proprio computer però può comportare limitazioni in termini di prestazioni e stabilità, oltre che a questioni di sicurezza legate all'esposizione diretta della propria rete. Inoltre, la lettura del QR-code sul dispositivo deve essere effettuata tramite l'app di Metamask, preventivamente installata.

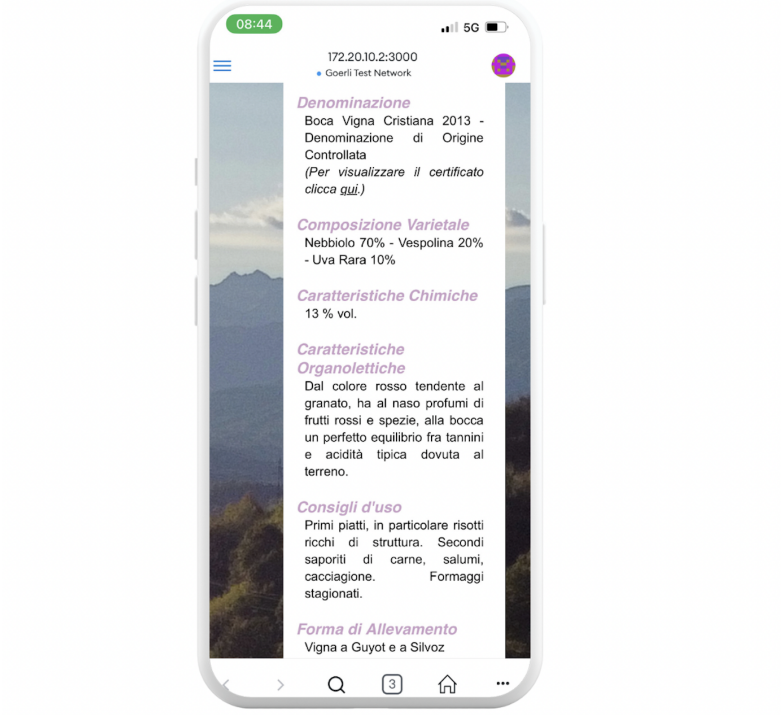

Figura 5.19: Visualizzazione su dispositivo mobile.

5.5 Risultati ottenuti

Di seguito viene riportata una Tabella raffigurante i risultati ottenuti dalla distribuzione dello smart contract tramite Truffle sulla rete di test Goerli. Una visualizzazione più completa dei risultati ottenuti da terminale digitando *truffle migrate –reset –network goerli* è disponibile in Appendice (A.4).

Come specificato nel paragrafo precedente, la rete è stata configurata con un Gas Price di 45Gwei e un Gas Limit di 15 milioni.

Gas Price (Gwei)	45.00
Time (min)	11.70
Gas Used	5251248.00
Cost (ETH)	0.24
Cost (Euro)	358.52

Tabella 5.2: Primo risultato della distribuzione dello smart contract sulla rete di test Goerli.

Tale impostazione è stata mantenuta per rispettare le previsioni sui prezzi del gas ottenuti tramite Etherscan, come mostrato in Figura 5.17. Il tempo della distribuzione dello smart contract sulla rete però risulta particolarmente elevato, così come il costo sostenuto. Si è dunque ritenuto opportuno effettuare più tentativi con l'obiettivo di comprendere se gli alti valori ottenuti nella Tabella 5.2 fossero dovuti al modo in cui lo smart contract era stato implementato o all'elevata congestione della rete nel momento in cui lo smart contract è stato distribuito.

In generale, il costo della distribuzione di uno smart contract sull rete Ethereum dipende da:

- *dimensione (in byte) e complessità dello smart contract*: il costo dipende dal

numero di righe di codice dello smart contract, poiché ogni riga di codice è un'operazione che richiede risorse computazionali per essere eseguita. Come regola generale, più linee di codice ci sono nel contratto smart, più alto è il costo di creazione;

- *gas*: il costo del deploy dello smart contract dipende dal gas necessario per eseguire le operazioni dello smart contract sulla blockchain. Il gas è una misura della quantità di risorse computazionali necessarie per eseguire un'operazione;

- *condizioni di mercato*: i costi di deploy possono variare in base alla domanda e all'offerta di risorse di calcolo sulla blockchain. In periodi di congestione della rete, i costi di deploy possono aumentare.

In sintesi, il costo e le tempistiche del deploy dello smart contract variano in base alla congestione della rete e, di conseguenza, al prezzo del gas. Infatti, quando ci sono molte transazioni che attendono di essere approvate sulla rete, il costo necessario per effettuare una transazione aumenta, influendo sul costo complessivo della distribuzione dello smart contract sulla rete. Il prezzo del gas è determinato dalla domanda e dall'offerta di gas sulla piattaforma Ethereum. Proprio per questo, occorre valutare attentamente lo stato di congestione della rete avvalendosi di tool come il gas tracker di Etherscan prima di effettuare il deploy del proprio contratto. Al contrario, quando la rete è poco congestionata, c'è meno domanda di gas, di conseguenza il prezzo per effettuare una transazione in tempi brevi diminuisce e il costo complessivo risulterà notevolmente più basso.

Per i motivi sopra citati, si è scelto di testare nuovamente gli smart contract in un momento in cui la rete è meno congestionata: l'obiettivo è verificare la possibilità di ottenere risultati migliori in termini di costi e tempi.

La prima operazione effettuata è stata monitorato lo stato della rete per mezzo di Etherscan; i nuovi tentativi sono stati effettuati quando le previsioni sul prezzo

del gas erano più basse. Sono stati realizzati cinque tentativi. Il Gas Price è stato impostato con valori che vanno da 30 a 5 Gwei. Questa scelta perché, al momento del deploy, il gas tracker di Etherscan riportava in media un valore di Gas Price per transazione compreso tra i 15 e i 20 Gwei. I risultati ottenuti sono riportati, in forma tabulare, di seguito:

Prezzo Gas (Gwei)	**Tempo impiegato (sec)**	**Costo (ETH)**	**Costo (Euro)**
30	13	0.1575822	269,25
20	15	0.10005056	169,44
15	23	0.0787911	133,44
10	37	0.0525274	88,96
5	570	0.0262637	44,88

Tabella 5.3: Secondo risultato della distribuzione dello smart contract sulla rete di test Goerli.

Si può notare come i risultati ottenuti siano significativamente migliori dei precedenti riguardo a tempi e costi. I grafici di seguito riportati dimostrano come, in generale, utilizzando un valore di gas price inferiore, il costo complessivo del deploy dello smart contract sulla rete sarà più basso ma, il tempo di elaborazione potrebbe risultare maggiore. Questa soluzione può essere una scelta vincente quando la rete non è molto congestionata, come nel caso in esame. Infatti, utilizzando un Gas Price di 5 Gwei, il costo risulta significamene più basso e il tempo di elaborazione è di 570 secondi, quindi circa 9 minuti.

Al contrario, utilizzando un gas price più alto, il costo complessivo del deploy dello smart contract è maggiore ma il tempo molto minore (di soli pochi secondi). É utile ricorre a questa soluzione quando la rete blockchain è molto congestionata e il tempo di elaborazione è un fattore critico.

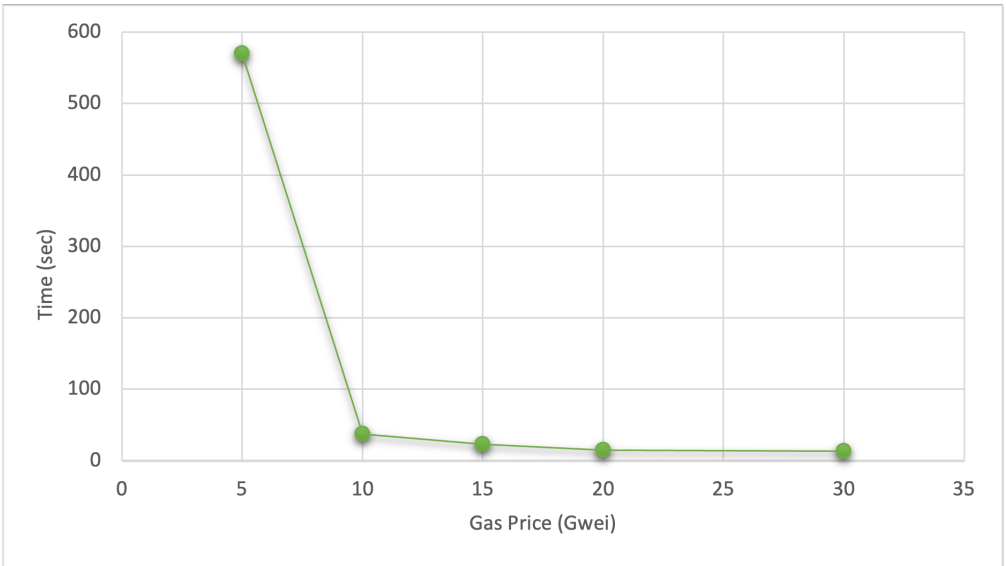

Figura 5.20: Andamento del tempo di elaborazione dello smart contract rispetto al Gas Price utilizzato.

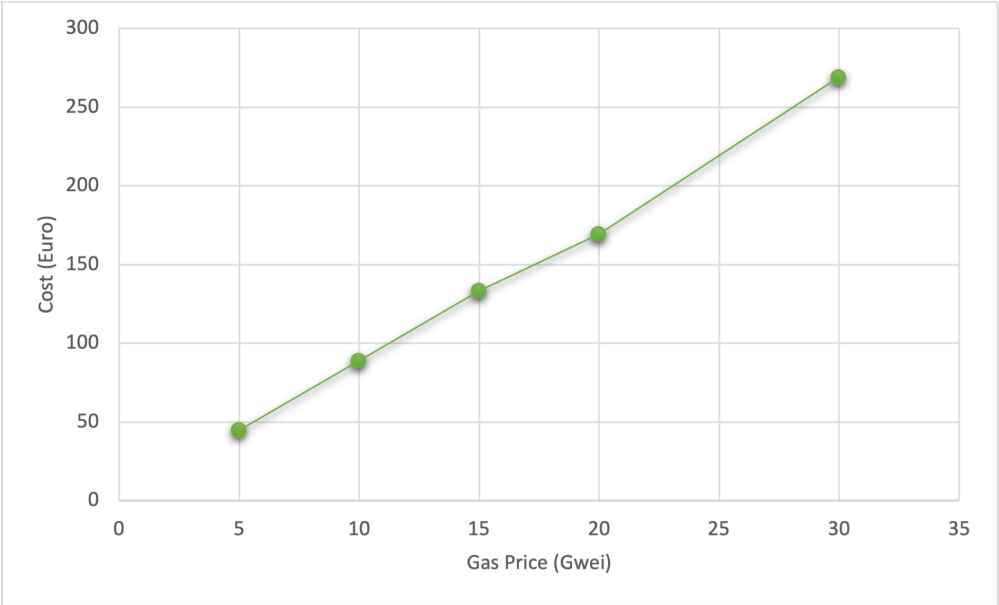

Figura 5.21: Andamento del costo dello smart contract rispetto al Gas Price utilizzato.

Nel caso in esame, non essendovi la necessità di tempi di elaborazione particolarmente brevi per la distribuzione dello smart contract sulla rete, se la rete risulta poco congestionata, conviene utilizzare un Gas Price basso in quanto il costo da sostenere sarà significativamente inferiore. Al costo necessario da sostenere per il deploy e lo storage dello smart contract sulla rete, va poi aggiunto quello per effettuare le singole transazioni.

Capitolo 6

Valutazioni e conclusioni

Alla fine di questo lavoro di tesi si può agevolmente confermare che nel settore agro-alimentare la tecnologia blockchain rappresenta lo strumento che le aziende possono utilizzare per consentire la tracciabilità del prodotto e certificare la qualità e sostenibilità dei processi produttivi. In modo da assecondare la crescente richiesta di trasparenza da parte dei consumatori. Il pilastro portante del lavoro di tesi è stato l'analisi dei requisiti per lo sviluppo di un'applicazione decentralizzata per un'azienda del settore vitivinicolo, la cantina "Podere ai Valloni". Individuate le richieste dell'azienda e definiti i requisiti, si è passati allo studio delle tecnologie da impiegare per lo sviluppo del prodotto e, infine, a un'accurata descrizione del prodotto stesso. É importante evidenziare che il prototipo sviluppato rappresenta solo una versione preliminare del prodotto finale, sviluppata principalmente per dimostrare la fattibilità dell'idea della start-up Sblockchain Project e ottenere un feedback iniziale dal cliente. Come tale, non comprende tutte le funzionalità e le caratteristiche richieste dalla specifica ma queste possono essere implementate in una fase successiva dello sviluppo.

6.1 Valutazione della dApp

Con il prototipo realizzato, la start-up Sblockchain Project ha la possibilità di avere una prima idea sul funzionamento del prodotto che sta sviluppando, testare le sue funzionalità e raccogliere feedback. Questo permette di identificare eventuali problemi o miglioramenti da apportare prima di procedere con la realizzazione del prodotto finale. Permette, inoltre, una più agevole comunicazione con i clienti e gli investitori, perché, consente di mostrare loro un'anteprima del prodotto e suscitare interesse.

Il prototipo proposto risponde all'esigenza dell'azienda vitivinicola di aumentare la fiducia dei consumatori, sia per quanto riguarda la qualità del prodotto sia nei confronti dell'azienda produttrice. Grazie al QR-code, il consumatore può accedere a tutte le informazioni, comprese quelle relative al contesto di produzione, ai dettagli sul processo produttivo e alle caratteristiche del prodotto. Questo, oltre a fornirgli un quadro più completo sullo stesso, gli permette di acquistare consapevolmente, prendere decisioni più informate, evitare il rischio di frode e scegliere prodotti di qualità più elevata. La dApp semplifica il processo di verifica dell'autenticità dei prodotti: il cliente sarà certo di star acquistando un prodotto originale e non una sua contraffazione. Inoltre, salvare dati e documenti su blockchain pubblica, sfruttando gli smart contract, garantisce l'immutabilità dei dati, quindi le informazioni saranno salvate in modo permanente, senza la possibilità di essere modificate, aumentando ulteriormente la sicurezza percepita dal consumatore.

6.2 Sviluppi Futuri

I possibili sviluppi futuri del prototipo realizzato potrebbero riguardare gli aspetti di seguito descritti:

- *migliorare l'esperienza utente*: in particolare, semplificare l'accesso alla dApp. Al momento, infatti, accedere alla pagina di visualizzazione richiede la scannerizzazione di un QR-code tramite Metamask, procedimento che può risultare non agevole per tutti i consumatori. Richiede, infatti, oltre ad aver scaricato l'app sul dispositivo mobile, di avere un proprio wallet Metamask e di essere collegati alla rete Ethereum principale.

  Per ovviare questo problema, sono state individuate due possibili soluzioni:

  1. ricorrere a tecniche di estrazione dati dagli smart contract differenti (come TheGraph) per rendere la lettura del codice e l'accesso alla pagina più semplice e intuitiva;
  2. invece di riportare sulla pagina web i dati estratti dallo smart contract, si potrebbe optare per riportare i codici delle transazioni sulla stessa. Gli hash di transazione sono una stringa alfanumerica lunga 64 caratteri che rappresenta univocamente una transazione sulla blockchain. Questi codici possono essere visualizzati come link cliccabili sulla pagina web, consentendo agli utenti di accedere direttamente alla transazione corrispondente sulla blockchain. Cliccando sul link, si verrà riportati sulla pagina Etherscan della transazione che ne include tutti i dettagli (costo, timestamp, ecc). Questo mostrerà all'utente l'effettiva esistenza della transazione e la veridicità dei dati presentati.

- *rendere disponibile la dApp in più lingue*: l'azienda vitivinicola esporta il proprio prodotto in altri stati Europei. Aggiungere la possibilità di visualizzare le informazioni in più lingue, preserva il Made In Italy all'estero e amplifica il numero di potenziali utenti della dApp;

- *ricorrere ad un servizio di hosting*: permetterebbe di rendere il sito web accessibile a tutti;

- *coinvolgere i fornitori*: permetterebbe di salvare su Blockchain tutta la filiera produttiva, così da rendere il processo ancora più trasparente. Questo richiederebbe l'aggiornamento degli smart contract. Inoltre, il sistema Blockchain potrebbe essere esteso anche ai pagamenti e il passaggio di proprietà di un prodotto potrebbe essere contrassegnato anche da un trasferimento di criptovaluta;

- *integrare con IoT*(come sensori o altri oggetti intelligenti): questi dispositivi possono fornire dati in tempo reale su posizione, stato, condizioni ambientali.

In conclusione, con la dApp realizzata, il consumatore potrà essere sicuro dell'origine, del processo produttivo e della qualità del prodotto acquistato mentre l'azienda vitivinicola potrà certificare in modo semplice e non falsificabile tutte le caratteristiche del prodotto. Il prototipo realizzato permette maggiore trasparenza e sicurezza dei dati rispetto ai metodi tradizionali, tempi e costi di sviluppo ridotti.

Appendice A

Appendice

Figura A.1: Numero di documenti con "Blockchain" come topic su WebOfScience: suddivisione per tipologia di documento.

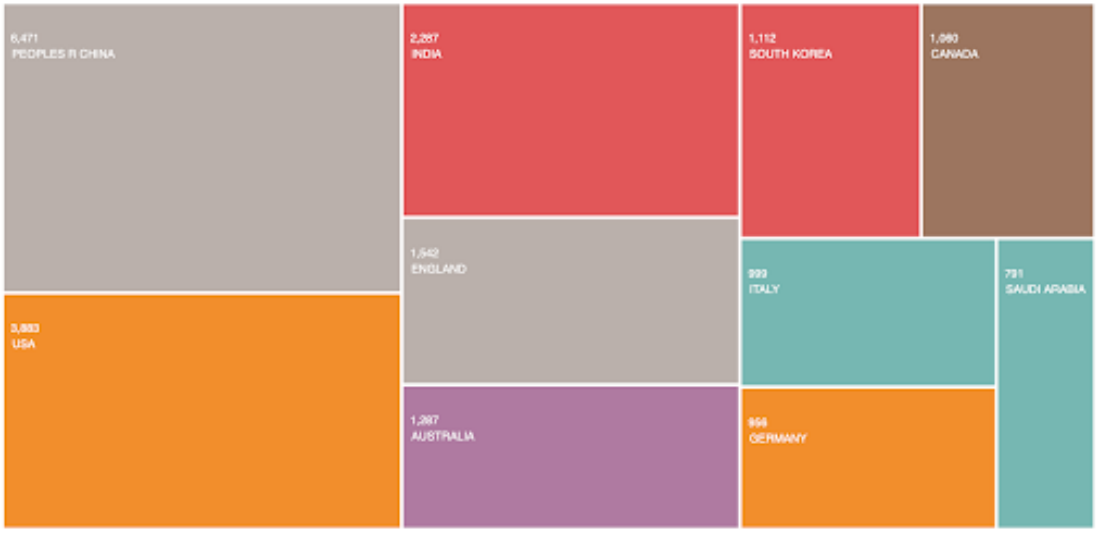

Figura A.2: Numero di documenti con "Blockchain" come topic su WebOfScience: suddivisione per area Geografica.

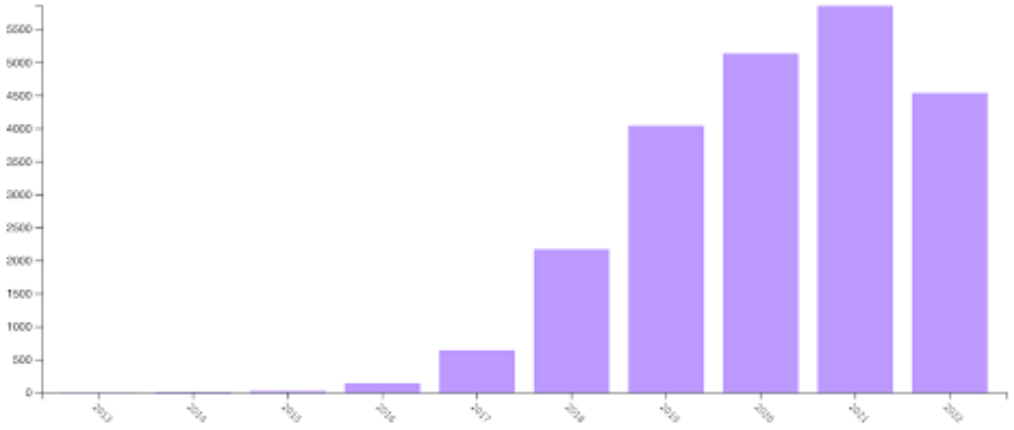

Figura A.3: Numero di documenti con "Blockchain" come topic su WebOfScience: suddivisione per anno.

```
Compiling your contracts...
===========================
> Everything is up to date, there is nothing to compile.


Starting migrations...
======================
> Network name:    'goerli'
> Network id:      5
> Block gas limit: 30000000 (0x1c9c380)


1_initial_migration.js
======================

   Replacing 'Migrations'
   ----------------------
   > transaction hash:    0x6deb4bc9fc3041f4c7416ac4559c00b8cf7c8cfe645b57b1adf4e29a73f5ac82
   > Blocks: 48           Seconds: 693
   > contract address:    0x55b34c2a4BD9A5632159234B4f852b386Bb07Cad
   > block number:        8532650
   > block timestamp:     1677013452
   > account:             0x9900963944FbB1a698F872810a8e212492F4B7cB
   > balance:             5.838986120014318666
   > gas used:            250154 (0x3d12a)
   > gas price:           45 gwei
   > value sent:          0 ETH
   > total cost:          0.01125693 ETH


   > Saving migration to chain.
   > Saving artifacts
   -------------------------------------
   > Total cost:          0.01125693 ETH


2_deploy_contract.js
====================

   Replacing 'MainContract'
   ------------------------
   > transaction hash:    0xc20f391155897e40308cdc00ee494926f50a4682782d6ba8b3e9ef213b5b6526
   > Blocks: 1            Seconds: 9
   > contract address:    0x834B97851aBc09041A08dfE3D613D338692de2d7
   > block number:        8532653
   > block timestamp:     1677013488
   > account:             0x9900963944FbB1a698F872810a8e212492F4B7cB
   > balance:             5.611870805014318666
   > gas used:            5001094 (0x4c4f86)
   > gas price:           45 gwei
   > value sent:          0 ETH
   > total cost:          0.22504923 ETH


   > Saving migration to chain.
   > Saving artifacts
   -------------------------------------
   > Total cost:          0.22504923 ETH


Summary
=======
> Total deployments:   2
> Final cost:          0.23630616 ETH
```

Figura A.4: Deploy dello smart contract sulla rete di Test Goerli, visualizzazione da terminale.

Ringraziamenti

Desidero ringraziare sentitamente la Prof.ssa Valentina Gatteschi, relatrice della mia tesi di laurea, per l' attenzione dedicatami sin dall'inizio, per avermi indirizzato e guidato nella stesura ed elaborazione di questo lavoro, per il suo costante incoraggiamento e supporto, per i suoi preziosi consigli e le esaustive e tempestive risposte alle mie richieste.

Desidero inoltre ringraziare Sblockchain Project per avermi accolto durante l'attività di tirocinio e di tesi. Sono grata per l'opportunità che mi è stata data di vivere l'ambiente della start-up: un contesto dinamico e stimolante. Ringrazio Francesco Reviglio Della Veneria e Andrea Luciano per avermi affiancato con la loro disponibilità e per avermi spronato e stimolato.

Infine, vorrei ringraziare la mia famiglia e i miei amici per il loro affetto, la loro pazienza e il loro costante supporto durante tutto il percorso di studi.

Grazie!

Bibliografia

[1] Satoshi Nakamoto. «Bitcoin: A Peer-to-Peer Electronic Cash System». In: (2008) (cit. alle pp. 11, 13).

[2] Stuart Haber e W. Scott Stornetta. «How To Time-Stamp a Digital Document». In: (1991) (cit. a p. 12).

[3] Roopika J. «Blockchain Technology: History, Concepts, and Applications». In: *International Research Journal of Engineering and Technology (IRJET)* (ott. 2020) (cit. alle pp. 12, 14).

[4] Moni Naor Cynthia Dwork. «Pricing via Processingnor Combatting Junk Mail». In: *Advances in Cryptology: CRYPTO 1992. Annual International Cryptography Conference, Santa Barbara, California, USA* (1992) (cit. a p. 12).

[5] Q.H. Kakarlapudi P.V.; Mahmoud. «A Systematic Review of Blockchain for Consent Management». In: (2021) (cit. a p. 13).

[6] Maria Johnsen. *Blockchain in Digital Marketing: A New Paradigm of Trust*. Mag. 2020 (cit. a p. 14).

[7] J. Yli-Huumo, D. Ko, S. Choi, S. Park e K. Smolander. «Where Is Current Research on Blockchain Technology?—A Systematic Review». In: (2016) (cit. a p. 16).

[8] L. Ghiro, F. Restuccia, S. D'Oro, S. Basagni, T. Melodia, L. Maccari e R. Lo Cignor. «What is a Blockchain? A Definition to Clarify the Role of the Blockchain in the Internet of Things». In: (feb. 2021) (cit. alle pp. 19, 22).

[9] Syed T. A, Alzahrani A. Jan S. e Siddiqui M. S. «A Comparative Analysis of Blockchain Architecture and Its Applications: Problems and Recommendations». In: *IEEE Access* (dic. 2019) (cit. a p. 19).

[10] M. Garriga, S. Dalla Palma, M. Arias, A. De Renzis, R. Pareschi e D.A. Tamburri. «Blockchain and Cryptocurrencies: a Classification and Comparison of Architecture Drivers». In: (lug. 2020) (cit. alle pp. 20, 54).

[11] *MIT Tecnology Review, Why Ethereum is switching to proof of stake and how it will work*. URL: `https://www.technologyreview.com/2022/03/04/1046636/ethereum-blockchain-proof-of-stake/` (cit. a p. 20).

[12] Hiren B. Patel Bela Shrimali. «Blockchain state-of-the-art: architecture, use cases, consensus, challenges and opportunities». In: *Journal of King Saud University – Computer and Information Sciences 34 (2022) 6793–6807* (2021) (cit. a p. 22).

[13] FAO. «Emerging opportunities for the application of blockchain in the agri-food Industry». In: (lug. 2020) (cit. a p. 23).

[14] Mauro Mandrioli Niccolò Patelli. «Blockchain technology and traceability in the agrifood industry». In: *Journal of Food Science* (2020) (cit. alle pp. 24, 33).

[15] Priyanka Arora1 e Ritu Makani. «Blockchain Technology and Its Applications: A Systematic Review of the Literature». In: *International Conference on Computational and Intelligent Data Science* () (cit. a p. 27).

[16] M.A.N. Agi e A.K. Jha. «Blockchain technology in the supply chain: An integrated theoretical perspective of organizational adoption». In: *Elsevier* (2022) (cit. a p. 28).

[17] Nikhil Vadgama e Paolo Tasca. «An Analysis of Blockchain Adoption in Supply Chains Between 2010 and 2020». In: *Centre for Blockchain Technologies, University College London, London, United Kingdom* (mar. 2021) (cit. alle pp. 28, 55).

[18] European Commission. «Food Traceability». In: (giu. 2007) (cit. a p. 29).

[19] FMI - The Food Industry Association e Label Insight. «Transparency Trends: Omnichannel Grocery Shopping from the Consumer Perspective». In: (2018) (cit. a p. 30).

[20] Vinay Singh e Sanjeev Kumar Sharma. «Application of blockchain technology in shaping the future of food industry based on transparency and consumer trust». In: (2018) (cit. alle pp. 30, 35).

[21] Antonucci F., Figorilli S., Costa C., F. Pallottino, Rasob L. e Menesattia P. «A review on blockchain applications in the agri-food sector». In: (apr. 2019) (cit. a p. 32).

[22] Violino S., Pallottino F., Sperandio G., Figorilli S., Antonucci F., Ioannoni V. e Costa C. «Are the innovative electronic labels for extra virgin olive oil sustainable, traceable and accepted by consumers?» In: (2019) (cit. a p. 33).

[23] Pierre Noizat. «Blockchain Electronic Vote». In: (dic. 2015) (cit. a p. 33).

[24] *Vino e frodi: in Oltrepò Pavese arresti per vino spacciato per Dop e Igp senza esserlo*. URL: `https://winenews.it/it/vino-e-frodi-in-oltrepo-pavese-arresti-per-vino-spacciato-per-dop-e-igp-senza-esserlo_408346/` (cit. a p. 35).

[25] Luzzani G.and Grandis E. e Capri E. Frey M. «Blockchain Technology in Wine Chain for Collecting andAddressing Sustainable Performance: An Exploratory Study». In: *Sustainability* (2021) (cit. a p. 35).

[26] *MyStory*. URL: `https://www.dnvgl.it/mystory/index.html` (cit. a p. 36).

[27] *Wine Blockchain*. URL: `https://www.ezlab.it/it/i-nostri-casi-di-studio/wine-blockchain/` (cit. a p. 36).

[28] *EY blockchain platform supports Blockchain Wine Pte. Ltd. to launch TATTOO Wine marketplace across Asia Pacific*. URL: `https://www.ey.com/en_gl/news/2019/11/ey-blockchain-platform-supports-blockchain-wine-pte-ltd-to-launch-tattoo-wine-marketplace-across-asia-pacific` (cit. a p. 37).

[29] Mediobanca. «Indagine sul settore vinicolo». In: (apr. 2019) (cit. a p. 39).

[30] *Introducing a16z crypto*. URL: `https://a16zcrypto.com/content/announcement/introducing-a16z-crypto/` (cit. a p. 40).

[31] Ömer Özgür Tanrıöver Ruhi Taş. «Building A Decentralized Application on the Ethereum Blockchain». In: *IEEE* (2019) (cit. a p. 40).

[32] Wee Lum Tan Kamanashis Biswas Vallipuram Muthukkumarasamy. «Blockchain based Wine Supply Chain Traceability System». In: *Future Technologies Conference (FTC)* (2017) (cit. a p. 43).

[33] Kim Fowler Chris Hersman. *Mission-Critical and Safety-Critical Systems Handbook*. 2010. Cap. 5, Best Practices in Spacecraft Development (cit. a p. 43).

[34] Hiren B. Patel Bela Shrimali. «Understanding the Requirement Engineering for Organization:The Challenges». In: *IEEE* (apr. 2012) (cit. a p. 44).

[35] Marco Torchiano Fulvio Corno. *Sistemi informativi aziendali - appunti per il corso, Versione 0.1.0*. Gen. 2021 (cit. alle pp. 49, 52).

[36] Roberto Tonelli Lodovica Marchesi Michele Marchesi. «ABCDE –agile block chain DApp engineering». In: *Elsevier* (2020) (cit. a p. 50).

[37] «Documentazione ufficiale Solidity». In: *URL: https://docs.soliditylang.org/en/latest/* () (cit. a p. 57).

[38] Soufiane Mezroui Ahmed El Oualkadi Aicha Bouichou. «An overview of Ethereum and Solidity vulnerabilities». In: *IEEE* (2020) (cit. a p. 57).

[39] Jakub Zakrzewski. «Towards verification of Ethereum smart contracts: a formalization of core of Solidity». In: () (cit. a p. 58).

[40] *Documentazione Node.js*. URL: `https://nodejs.org/it/docs/` (cit. a p. 59).

[41] *Documentazione Truffle*. URL: `https://trufflesuite.com/docs/` (cit. a p. 59).

[42] *Documentazione Ganache*. URL: `https://trufflesuite.com/docs/ganache/` (cit. a p. 60).

[43] *Web3.js - Ethereum JavaScript API*. URL: `https://web3js.readthedocs.io/en/v1.8.2/index.html` (cit. a p. 62).

[44] *Documentazione IPFS*. URL: `https://docs.ipfs.tech/` (cit. a p. 62).

[45] *Pinata Docs*. URL: `https://docs.pinata.cloud/` (cit. a p. 63).

[46] *Documentazione Metamask*. URL: `https://docs.metamask.io/guide/` (cit. a p. 65).

[47] *Goerli Testnet*. URL: `https://goerli.net/` (cit. a p. 67).

[48] *Documentazione Infura*. URL: `https://docs.infura.io/infura/` (cit. a p. 68).

[49] Dr. A. Rengarajan Soham Toraskar. «Research paper on Storing Documents on Blockchain 2021-2022». In: (apr. 2022) (cit. a p. 70).

[50] K.L. Guadalupe-Gallardo S.I. Soto-Ortiz P.J. Salazar-Pérez. «Development of a DAPP (decentralized application) on Rinkeby Network for the registration of Sensors of an Internet of Things (IoT) Environment». In: *IEEE* (2022) (cit. a p. 79).