ALMA MATER STUDIORUM
UNIVERSITÀ DI BOLOGNA

SCUOLA DI SCIENZE

Corso di Laurea in Informatica

SMART INSURANCE DECENTRALIZZATE:

PROGETTAZIONE E SVILUPPO DI UN SISTEMA BASATO SU BLOCKCHAIN PER LA GESTIONE AUTOMATICA DI POLIZZE PARAMETRICHE

Relatore:
FEDERICO MONTORI

Presentata da:
MARCO GUIDI

Correlatore:
LORENZO GIGLI

Sessione II
2024-25

Abstract

Il settore assicurativo sta affrontando una trasformazione profonda, trainata dall'introduzione di tecnologie decentralizzate e dall'uso di contratti intelligenti. Tuttavia, gran parte dei modelli di *smart insurance* esistenti si limita ad automatizzare la fase contrattuale, senza affrontare in modo efficace due criticità centrali: la qualità e l'affidabilità dei dati esterni, e la sostenibilità economica del sistema.

Questa tesi propone la progettazione e lo sviluppo di un ecosistema assicurativo decentralizzato basato su *smart contracts* e sull'integrazione con il sistema **Zonia**, un'infrastruttura di oracoli e validazione dati fondata anch'essa su blockchain e smart contracts. Tale architettura consente di acquisire e verificare in modo trasparente dati provenienti da sensori IoT, garantendo che solo informazioni certificate possano attivare le clausole assicurative.

Il lavoro combina quindi la componente tecnica — la realizzazione di una piattaforma mobile basata su *React Native* ed Ethers.js e la scrittura e deploy di smart contracts dedicati — con l'analisi economica delle *gas fee*, volta a valutare la sostenibilità del modello e a definire una linea guida per l'introduzione di **commissioni dinamiche**. I risultati dei test confermano la stabilità dell'architettura, la correttezza logica dei contratti e la fattibilità economica del sistema.

Nel suo complesso, la tesi dimostra come l'integrazione tra blockchain, IoT e oracoli basati su smart contracts possa rendere il modello assicurativo più efficiente, verificabile e autosufficiente, aprendo la strada a una nuova generazione di polizze parametriche trasparenti e decentralizzate.

Indice

Elenco delle figure

Introduzione

Negli ultimi anni, il settore assicurativo ha iniziato a confrontarsi con una crescente pressione verso la digitalizzazione e la trasparenza. Tuttavia, nonostante l'adozione di soluzioni informatiche avanzate, i processi fondamentali restano ancora centralizzati e dipendenti dall'intervento umano, con conseguenti inefficienze, tempi di gestione lunghi e margini di errore elevati.

L'avvento delle tecnologie blockchain e dei *smart contracts* ha introdotto un nuovo paradigma operativo: l'automazione dei rapporti fiduciari attraverso la programmazione delle clausole contrattuali. Nel contesto assicurativo, ciò ha dato origine al concetto di *smart insurance*, in cui la liquidazione di un sinistro può avvenire in modo automatico e verificabile al verificarsi di determinate condizioni. Tuttavia, i modelli attuali presentano ancora due criticità principali:

- l'affidabilità dei dati utilizzati per attivare le clausole contrattuali, spesso provenienti da oracoli centralizzati o non verificabili;
- l'assenza di una strategia sostenibile per la copertura dei costi di rete e la gestione economica del sistema.

Questa tesi nasce con l'obiettivo di affrontare tali sfide proponendo un modello di **smart insurance decentralizzata** che unisce contratti intelligenti, sensori IoT e oracoli blockchain, convalidati attraverso l'infrastruttura **Zonia**. Quest'ultima, fondata anch'essa su smart contracts, garantisce che i dati provenienti dal mondo reale siano certificati e tracciabili, permettendo una gestione delle polizze realmente automatizzata e affidabile.

L'applicazione sviluppata, **Trust App**, rappresenta la dimostrazione pratica di questa architettura: un ecosistema in cui utenti e compagnie possono interagire direttamente tramite blockchain, gestendo la vita di una polizza senza intermediari e con la sicurezza garantita da meccanismi crittografici.

Oltre alla validazione tecnica, la tesi analizza anche la dimensione economica del sistema, valutando le *gas fee* delle transazioni e proponendo un modello di **commissioni dinamiche** che consenta al gestore di mantenere la sostenibilità finanziaria del servizio senza compromettere l'accessibilità per l'utente finale.

In sintesi, questo lavoro si pone come contributo alla costruzione di un modello assicurativo innovativo, basato su trasparenza, automazione e sostenibilità economica.

Capitolo 1

Smart Insurance e Innovazioni Tecnologiche: Index-Based Insurance, IoT e Smart Contracts

1.1 Introduzione

Le assicurazioni tradizionali si basano su processi centralizzati e manuali, in cui la valutazione del rischio, la gestione dei sinistri e la determinazione dei premi dipendono da procedure complesse, spesso opache e soggette a discrezionalità. Questo approccio comporta tempi di gestione elevati, costi significativi e talvolta insoddisfazione degli assicurati. Negli ultimi anni, grazie all'evoluzione delle tecnologie digitali e all'adozione di infrastrutture distribuite, è emerso il paradigma delle *smart insurance* (SI), che mira a rendere il settore assicurativo più efficiente, trasparente e personalizzato [1].

Le smart insurance si fondano su tre pilastri principali: la raccolta di dati in tempo reale tramite dispositivi IoT, l'automazione contrattuale tramite *smart contracts* e l'immutabilità garantita dalle blockchain. Questi elementi consentono di monitorare continuamente i rischi, regolare automaticamente

i premi e liquidare sinistri senza intervento manuale, riducendo i tempi e migliorando la fiducia degli utenti. Il valore aggiunto di questo modello emerge soprattutto nei settori dove i rischi sono variabili, rapidi e difficili da monitorare, come la mobilità e l'agricoltura.

Inoltre, l'adozione di meccanismi parametrici, in cui il pagamento dell'indennizzo è legato al verificarsi di eventi misurabili, riduce le controversie e accelera la gestione dei sinistri. La disponibilità di dati affidabili, raccolti in maniera granulare tramite sensori ambientali, diventa così un fattore cruciale per garantire la correttezza e la trasparenza dei processi, rafforzando la fiducia degli assicurati e aprendo la strada a un ecosistema assicurativo più resiliente e adattabile.

1.2 Index-Based Insurance

Le *index-based insurance* (IBI), note anche come assicurazioni parametriche, rappresentano un approccio innovativo nel contesto delle smart insurance. In queste polizze, l'erogazione dei pagamenti non dipende dalla verifica diretta del danno subito dall'assicurato, ma dal superamento di soglie prestabilite di specifici indici misurabili, come precipitazioni, temperatura o rese agricole [2]. Questo modello consente di ridurre drasticamente i costi di gestione e i tempi di liquidazione, eliminando la necessità di sopralluoghi e valutazioni individuali [3].

Gli indici utilizzati possono essere di diversa natura: meteorologici (pioggia, temperatura, vento), ambientali (inquinamento, qualità dell'aria, livello di acqua), agricoli (resa stimata del raccolto tramite immagini satellitari) o industriali (vibrazioni, temperature in impianti). La progettazione di indici accurati richiede modelli statistici predittivi, raccolta dati storica e tecniche di remote sensing, al fine di ridurre il *basis risk*, ovvero il rischio che l'indice scelto non rifletta correttamente il danno reale [4].

Un'ulteriore sfida emerge nel contesto delle assicurazioni parametriche: il *moral hazard*. Il *moral hazard* indica il comportamento opportunistico

dell'assicurato, che può modificare le proprie azioni sapendo di essere coperto da una polizza. Nei sistemi basati su indici verificabili tramite sensori IoT e smart contracts, questo rischio può essere significativamente mitigato, in quanto i pagamenti dipendono da dati oggettivi esterni, e non dalle dichiarazioni del singolo [5].

In sintesi, le IBI permettono di combinare l'efficienza dei processi automatizzati con la solidità dei dati derivanti da sensori e infrastrutture blockchain, creando polizze più trasparenti, rapide e scalabili, capaci di coprire rischi in aree o settori dove le assicurazioni tradizionali risulterebbero inefficaci o troppo costose.

1.3 Casi d'uso delle assicurazioni parametriche

Le assicurazioni parametriche sono state inizialmente adottate nei contesti agricoli e climatici, dove la variabilità dei rischi ambientali è elevata e difficile da controllare. In queste situazioni, l'attivazione automatica di compensazioni basata su indici oggettivi riduce la complessità amministrativa e i conflitti sui sinistri.

In Paesi in via di sviluppo, le IBI aiutano le comunità rurali a migliorare la resilienza economica contro eventi meteorologici estremi e a pianificare investimenti agricoli più sicuri [3]. La disponibilità di dati storici e in tempo reale consente alle compagnie di offrire coperture più affidabili e di calibrare correttamente i premi. In contesti sviluppati, come Stati Uniti e Australia, l'integrazione di dati satellitari e modelli predittivi migliora la precisione degli indici, riducendo il *basis risk* e aumentando la trasparenza dei processi assicurativi [6].

Recentemente, si stanno esplorando applicazioni in ambito urbano e industriale, come la copertura automatica contro ondate di calore, inquinamento atmosferico o blackout energetici, dove l'IoT fornisce misurazioni granulari e continue dei parametri ambientali. Tali informazioni vengono poi utilizzate

come input per smart contracts che erogano i pagamenti automaticamente, migliorando tempi di liquidazione, trasparenza e tracciabilità [7].

1.4 Smart Contracts e Blockchain per le Smart Insurance

Gli smart contracts sono protocolli auto-eseguibili che permettono di automatizzare l'esecuzione di clausole contrattuali quando determinate condizioni sono soddisfatte [8]. Integrati con la blockchain, offrono immutabilità, trasparenza e tracciabilità, aspetti fondamentali in un settore come quello assicurativo, dove la fiducia tra le parti è essenziale [9].

Grazie agli smart contracts, l'intero ciclo assicurativo può essere automatizzato: dalla verifica di eventi predeterminati (ad esempio superamento di soglie ambientali) alla liquidazione dei sinistri, fino alla registrazione delle transazioni in modo permanente. Questa automazione riduce drasticamente tempi e costi, minimizzando errori umani e margini di discrezionalità.

Sono già presenti numerosi casi concreti. Etherisc ha sviluppato soluzioni decentralizzate per assicurazioni su voli aerei e colture agricole, dove il pagamento automatico si attiva al verificarsi di condizioni specifiche registrate da fonti affidabili [9].

I vantaggi concreti degli smart contract sviluppati includono:

- **Trasparenza:** tutte le condizioni contrattuali e le transazioni sono visibili e verificabili.
- **Riduzione delle frodi:** la registrazione immutabile dei dati limita manipolazioni e false dichiarazioni.
- **Efficienza operativa:** le polizze possono essere gestite con tempi di liquidazione quasi istantanei.
- **Scalabilità:** è possibile gestire simultaneamente un elevato numero di polizze senza aumentare proporzionalmente i costi.

Tuttavia, esistono anche criticità: errori nel codice del contratto, vulnerabilità informatiche o malfunzionamenti dei dati in input possono compromettere l'intero processo. Per mitigare questi rischi, si ricorre a:

- audit del codice e test approfonditi degli smart contracts;
- utilizzo di oracoli multipli e ridondanti per i dati esterni;
- aggiornamenti controllati e versioning dei contratti, mantenendo la tracciabilità delle modifiche.

L'integrazione con IoT amplia ulteriormente le possibilità: sensori distribuiti raccolgono dati in tempo reale, inviandoli direttamente agli smart contracts tramite oracoli sicuri. In questo modo, eventi come piogge estreme, livelli di inquinamento o variazioni di temperatura possono attivare automaticamente polizze parametriche, creando un ecosistema assicurativo dinamico, affidabile e altamente interconnesso.

Infine, l'adozione di smart contracts e blockchain favorisce anche la creazione di modelli di assicurazione collaborativi, in cui più compagnie o comunità possono condividere rischi e dati in modo trasparente, aprendo la strada a nuovi servizi assicurativi decentralizzati e innovativi.

1.5 IoT e raccolta centralizzata dei dati

La sensoristica IoT rappresenta la base per le smart insurance, consentendo la raccolta di dati ambientali affidabili e in tempo reale. Sensori distribuiti possono monitorare parametri quali temperatura, umidità, livelli di inquinamento, vibrazioni o altezza dell'acqua, fornendo informazioni precise e granulari. Questi dati vengono poi utilizzati come input per i contratti assicurativi automatizzati. L'integrazione con blockchain garantisce la provenienza dei dati, prevenendo manipolazioni e assicurando l'integrità delle misurazioni [10].

L'affidabilità dei dati IoT rimane una sfida: occorrono standard di sicurezza, sistemi di verifica e oracoli decentralizzati per assicurare la qualità

dei dati inviati agli smart contracts. L'uso di piattaforme centralizzate come Zonia [11] permette di consolidare e validare i dati raccolti da diverse fonti, assicurando la tracciabilità e riducendo errori o manomissioni. Questo approccio favorisce non solo la tempestività delle liquidazioni assicurative, ma anche lo sviluppo di nuovi servizi basati sull'analisi dei trend ambientali e comportamentali.

Inoltre, la disponibilità di dati continui e dettagliati consente di ottimizzare la progettazione degli indici parametrici, riducendo il *basis risk* e aumentando la precisione dei pagamenti automatici. La combinazione di sensori IoT, piattaforme centralizzate e smart contracts costituisce quindi un'infrastruttura robusta e scalabile, in grado di supportare modelli assicurativi innovativi e affidabili, capaci di adattarsi rapidamente a scenari mutevoli e di rispondere a nuovi tipi di rischio ambientale.

1.6 Aspetti normativi e prospettive

L'adozione delle smart insurance incontra ostacoli normativi. La validità legale degli smart contracts varia tra ordinamenti e resta aperta la questione della responsabilità in caso di malfunzionamenti [12]. La gestione dei dati IoT solleva inoltre problemi di privacy e di conformità a regolamenti come il GDPR.

Le prospettive future puntano all'integrazione con l'intelligenza artificiale per analisi predittive, allo sviluppo di oracoli affidabili e alla definizione di standard internazionali condivisi. La convergenza di blockchain, IoT e smart contracts promette un settore assicurativo più efficiente, trasparente e inclusivo.

1.7 Motivazioni

Il settore assicurativo, pur essendo uno dei più regolamentati, rimane fortemente dipendente da processi centralizzati e manuali, con tempi di gestione

elevati e scarsa trasparenza. Le soluzioni di *smart insurance* basate su blockchain e smart contracts hanno introdotto automazione e tracciabilità, ma gran parte dei modelli esistenti si limita a digitalizzare la fase contrattuale, trascurando la qualità dei dati e la gestione completa del ciclo di vita delle polizze.

Questa tesi nasce per colmare proprio tale lacuna, proponendo un modello di **index-based smart insurance** che integra in modo nativo il sistema **Zonia** per la validazione decentralizzata dei dati provenienti da sensori IoT. Grazie a questa integrazione, solo informazioni certificate e verificabili vengono trasmesse agli smart contracts, riducendo drasticamente il rischio di falsificazioni o incongruenze tra eventi reali e registrazioni on-chain. La blockchain diventa così non solo un registro di transazioni, ma una vera piattaforma di esecuzione basata su dati autenticati.

Un secondo elemento distintivo del modello riguarda la **gestione temporale e dinamica delle polizze**, che introduce logiche di scadenza automatiche, migliorando la trasparenza e la gestione del rischio da parte delle compagnie assicurative. Inoltre, la proposta include una prima linea guida per l'introduzione di **commissioni dinamiche**, concepite per bilanciare la sostenibilità economica del sistema con l'accessibilità per gli utenti.

Dal punto di vista tecnico, l'applicazione sviluppata dimostra la fattibilità di un ecosistema assicurativo decentralizzato, sicuro e utilizzabile, capace di astrarre la complessità della blockchain attraverso un'interfaccia intuitiva. Il contributo complessivo della tesi risiede quindi nella creazione di un modello architetturale che unisce **affidabilità dei dati**, **automazione completa del ciclo di vita delle polizze** e **sostenibilità economica**, ponendo le basi per una nuova generazione di assicurazioni parametriche realmente trasparenti e verificabili.

Capitolo 2

Architettura del Sistema

2.1 Introduzione all'Architettura

L'architettura proposta in questa tesi si fonda sull'integrazione di tre componenti tecnologiche fondamentali — IoT, blockchain e smart contracts — con l'obiettivo di automatizzare e rendere trasparente la gestione delle polizze assicurative parametriche. La mia applicazione funge da elemento di raccordo tra questi domini, fornendo un'interfaccia intuitiva per la creazione, la consultazione e il monitoraggio delle smart insurance.

A differenza dei sistemi tradizionali, che richiedono procedure manuali e centralizzate per la verifica dei sinistri, il modello presentato qui adotta un approccio completamente automatizzato. Le informazioni ambientali raccolte dai sensori IoT vengono validate e certificate da Zonia [11].

L'architettura mantiene una chiara separazione dei ruoli e delle responsabilità, garantendo modularità, affidabilità e scalabilità. Ciò consente di adattare il sistema a diversi scenari assicurativi e di estenderlo facilmente ad altre tipologie di dati sensoriali o di modelli contrattuali.

2.2 Schema architetturale del sistema

La Figura 2.1 illustra la struttura complessiva del sistema di smart insurance e il flusso dei dati tra i vari componenti. Il diagramma segue un modello lineare, evidenziando le interconnessioni principali tra sensori IoT, Zonia, blockchain e applicazione.

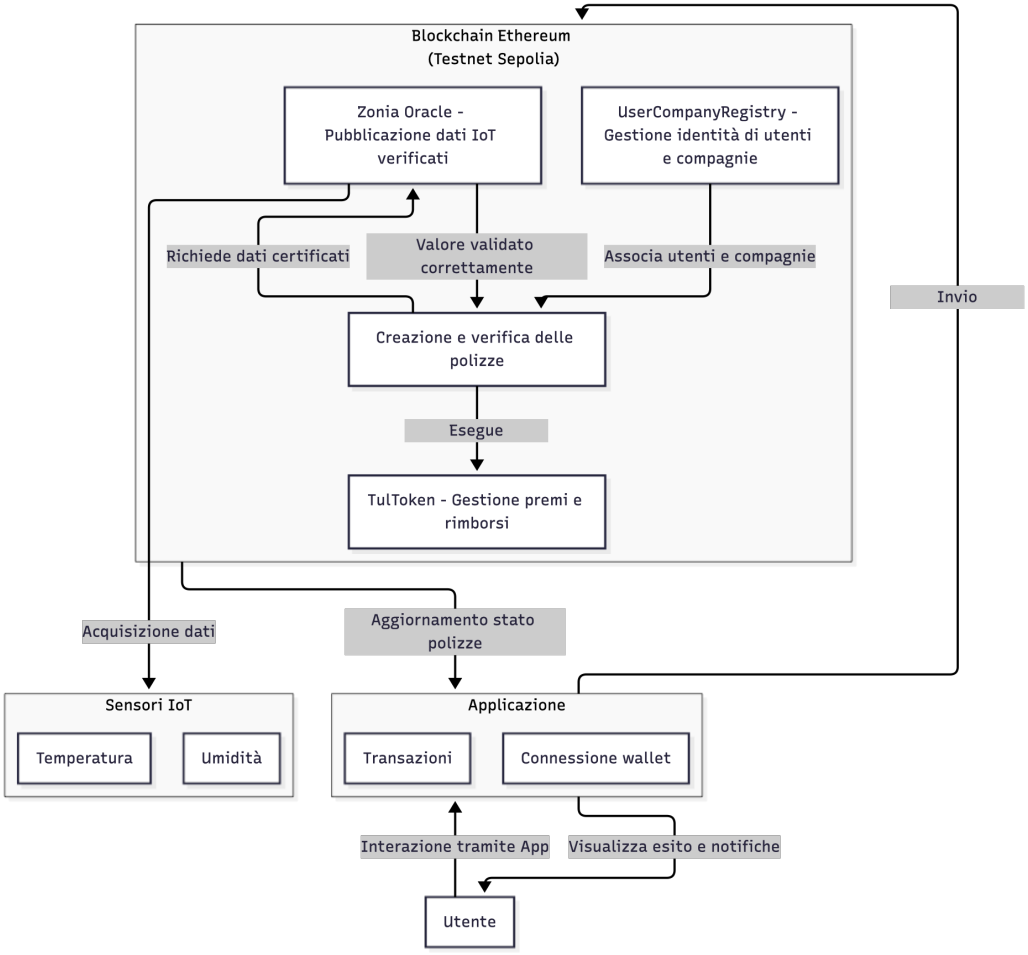

Figura 2.1: Architettura generale del sistema di Smart Insurance.

Nel diagramma, la componente applicativa — sviluppata nell'ambito di questa tesi — è evidenziata per mostrare il suo ruolo centrale di orchestrazione tra i vari livelli tecnologici.

2.3 Componenti principali del sistema

L'ecosistema sviluppato è costituito da tre livelli principali, ognuno con un ruolo specifico nel ciclo di vita della polizza.

2.3.1 Zonia Oracle System

Zonia rappresenta il livello di interfaccia con il mondo fisico, responsabile della raccolta, validazione e distribuzione dei dati provenienti dai sensori IoT. Il sistema riceve misurazioni ambientali e le processa per garantirne la qualità, l'autenticità e la tracciabilità.

La sua funzione di "oracolo affidabile" lo rende un punto chiave dell'architettura: solo i dati certificati dagli smart contracts di Zonia vengono riconosciuti dalle polizze, evitando così che valori errati o manipolati possano innescare pagamenti non dovuti. L'approccio proposto da Zonia consente inoltre di ridurre il rischio di discrepanze tra dati raccolti localmente e informazioni trasmesse alla blockchain, assicurando coerenza e integrità lungo tutto il processo.

2.3.2 Smart Contracts e Blockchain

Il secondo livello è rappresentato dagli *smart contracts* utilizzati, implementati su una rete blockchain pubblica. Ogni contratto contiene le regole e le soglie parametriche che definiscono la polizza assicurativa. Quando i dati provenienti da Zonia soddisfano le condizioni stabilite, il contratto viene eseguito, attivando la liquidazione del sinistro.

Oltre alle condizioni parametriche basate su dati ambientali, ogni polizza include anche una **scadenza temporale**. La scadenza rappresenta un elemento di controllo gestionale: una volta superato il periodo di validità della polizza, la compagnia richiede al contratto la chiusura della copertura. Questo meccanismo permette alla compagnia di riscattare le polizze non attivate e di aggiornare il portafoglio assicurativo in modo trasparente e verificabile.

L'uso della blockchain garantisce tracciabilità e immutabilità: ogni evento (creazione, aggiornamento, liquidazione o scadenza) è registrato in modo permanente. Ciò elimina la necessità di intermediari e favorisce la fiducia tra compagnia e assicurato, poiché tutti i processi possono essere verificati pubblicamente.

2.3.3 Applicazione Smart Insurance

L'applicazione rappresenta la componente centrale del sistema e il principale contributo progettuale di questa tesi. Essa funge da ponte tra l'utente e l'infrastruttura sottostante, consentendo di:

- creare nuove polizze parametriche, specificando i parametri ambientali e la durata;
- visualizzare in tempo reale i dati validati provenienti da Zonia;
- monitorare lo stato delle polizze attive, comprese eventuali liquidazioni o scadenze;
- fornire aggiornamenti sul ciclo di vita dei contratti.

L'applicazione non interagisce direttamente con i dispositivi IoT, ma riceve i dati già certificati da Zonia.

2.4 Zonia Oracle System

Zonia è un'infrastruttura decentralizzata per la raccolta, validazione e attestazione di dati provenienti da sorgenti fisiche o digitali, concepita per risolvere uno dei limiti strutturali più noti delle architetture basate su *smart contracts*: l'assenza di accesso nativo a informazioni esterne alla blockchain. A differenza degli oracoli tradizionali, che si limitano a fornire un flusso di dati esterno, Zonia introduce un modello di attestazione trasparente e verificabile, interamente basato su *smart contracts* e meccanismi di reputazione

decentralizzati, garantendo l'autenticità dei dati in ogni fase del loro ciclo di vita.

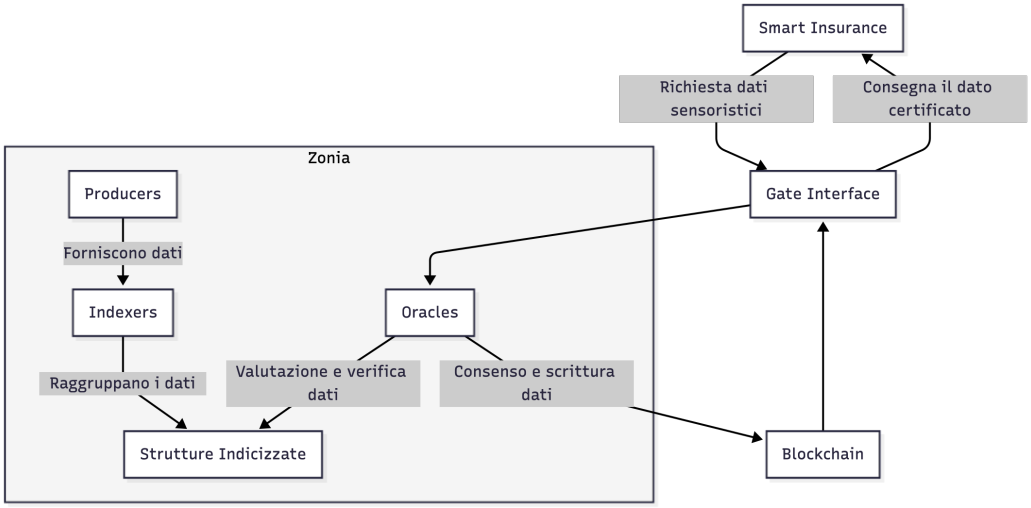

Figura 2.2: Immagine architetturale e integrazione di Zonia.

2.4.1 Struttura e ruoli principali

L'architettura di Zonia si articola su più livelli cooperativi, ciascuno con funzioni e responsabilità ben definite:

- **Producers:** rappresentano il punto di ingresso del sistema. Sono dispositivi o nodi autorizzati che raccolgono dati dai sensori IoT (ad esempio temperatura, umidità, precipitazioni) e li trasmettono al network Zonia, in cambio di token di ricompensa. Ogni producer è identificato e firmatario dei dati inviati, garantendo autenticità e tracciabilità.

- **Indexers:** ricevono i dati dai producers e li organizzano in strutture indicizzate, ottimizzate per la ricerca e l'accesso da parte degli oracoli. Questa fase intermedia riduce la latenza delle query e permette di ricostruire serie temporali o dataset geografici coerenti, fondamentali nei contesti assicurativi parametrici.

- **Oracoli:** sono i nodi che svolgono la funzione di verifica e validazione. Essi analizzano i dati provenienti dagli indexers, controllano la firma digitale dei producers, valutano eventuali incongruenze e redigono un attestato di veridicità. La validazione avviene in modo decentralizzato, tramite consenso tra più oracoli, riducendo la possibilità di errore o manipolazione. Gli oracoli, dopo aver validato le informazioni, le compattano in un unico dato e lo scrivono su chain. La blockchain funge da registro pubblico e immutabile, contenente le evidenze crittografiche che certificano l'origine e la validità dei dati.

- **Gate Interface:** rappresenta l'elemento di connessione tra Zonia e i sistemi esterni. È il livello con cui il sistema proposto interagisce direttamente: riceve richieste di validazione provenienti da smart contracts esterni (come quelli della *Smart Insurance*), inoltra la richiesta al network Zonia e restituisce un dato già attestato. Attraverso questa interfaccia, la blockchain principale non deve comunicare direttamente con i sensori, ma riceve esclusivamente valori certificati, riducendo l'esposizione a fonti non verificate.

2.4.2 Funzionamento del ciclo dei dati

Il ciclo informativo di Zonia può essere descritto in cinque fasi principali:

1. I **producers** raccolgono i dati dai sensori e li firmano crittograficamente.

2. Gli **indexers** aggregano i dati ricevuti, li ordinano temporalmente e li preparano per la fase di verifica.

3. Gli **oracoli** validano i dati confrontandoli con fonti ridondanti o parametri di coerenza.

4. Gli oracoli certificano l'esito della verifica tramite uno *smart contract* di attestazione, dove i dati vengono registrati on-chain.

5. La **Gate Interface** risponde alle richieste provenienti da applicazioni esterne, fornendo il dato certificato e il relativo attestato di validità.

2.4.3 Vantaggi architetturali e motivazioni della scelta

L'integrazione di Zonia nel sistema proposto non è stata casuale. Rispetto a soluzioni più diffuse come Chainlink o Band Protocol, Zonia offre diversi vantaggi chiave:

1. **Architettura completamente on-chain:** la maggior parte degli oracoli tradizionali affida la validazione dei dati a nodi off-chain, limitandosi a riportare il risultato in blockchain. Zonia, invece, effettua la validazione e l'attestazione direttamente tramite smart contracts, garantendo auditabilità completa del processo.

2. **Meccanismo di reputazione decentralizzato:** ogni producer e oracolo è associato a un punteggio reputazionale, aggiornato dinamicamente in base all'affidabilità delle attestazioni fornite. Questo modello incentiva il comportamento corretto dei partecipanti e penalizza chi fornisce dati errati o fraudolenti.

3. **Interoperabilità e modularità:** Zonia è progettato per essere agnostico rispetto al tipo di dato o alla blockchain di destinazione. Può alimentare diversi sistemi, inclusi contratti assicurativi parametrici, piattaforme di prestito o protocolli DeFi. La sua architettura modulare lo rende facilmente estendibile a nuovi domini applicativi.

4. **Sorgenti dati nascoste:** le fonti di dati utilizzate all'interno di Zonia non sono direttamente esposte ai client esterni. Gli utenti o i contratti che richiedono un'informazione non possono selezionare manualmente la sorgente da cui essa viene prelevata. Al contrario, Zonia interroga automaticamente più fonti di dati — scelte in base alla compatibilità semantica e geografica con la richiesta — e combina i risultati attraverso un algoritmo di composizione decentralizzato. Questo approccio

garantisce che il valore finale restituito sia aggregato e verificabile, riducendo la possibilità di manipolazione o bias derivante da una singola sorgente di dati.

2.4.4 Ruolo nel sistema Smart Insurance

Nel contesto del sistema proposto in questa tesi, Zonia svolge il ruolo cruciale di "ponte di fiducia" tra il mondo fisico e quello digitale. La **Gate Interface** è il punto in cui lo smart contract di ogni polizza interagisce con Zonia: attraverso una chiamata di verifica, richiede i dati ambientali (ad esempio temperatura o precipitazioni) relativi a una determinata area geografica. Una volta ottenuto il valore attestato, il contratto della polizza valuta se la soglia parametrica definita è stata superata e, in caso affermativo, attiva automaticamente la procedura di rimborso.

L'utilizzo di Zonia garantisce che il processo decisionale del contratto assicurativo si basi su informazioni affidabili, immutabili e verificabili da chiunque. Inoltre, il modello di attestazione riduce i rischi di frode o discrepanza tra evento reale e dato registrato, elevando la trasparenza complessiva del sistema.

2.4.5 Considerazioni finali

Zonia rappresenta una delle soluzioni oracolari più avanzate attualmente disponibili per contesti ad alta sensibilità dei dati, come quello assicurativo. La sua architettura distribuita e on-chain non solo risolve il problema della fiducia nei dati esterni, ma lo fa in modo verificabile e decentralizzato. Questa caratteristica, combinata alla compatibilità con sistemi esterni tramite la *Gate Interface*, rende Zonia la scelta ideale per l'infrastruttura proposta in questa tesi, consentendo di costruire un modello di smart insurance realmente autonomo, affidabile e sostenibile.

2.5 Flusso logico del sistema

Il flusso logico che regola il funzionamento dell'intero ecosistema può essere suddiviso in cinque fasi principali:

1. **Creazione della polizza:** la compagnia assicurativa definisce una nuova smart insurance, indicando i parametri ambientali di riferimento e la durata della copertura, oltre che valori di premio e risarcimento.

2. **Attivazione della polizza:** l'utente visualizza la polizza creata dalla compagnie ed effettua la transazione di pagamento del premio stabilito, la polizza diventa ora attiva e si può verificare.

3. **Raccolta e validazione dei dati:** su richiesta da parte dell'utente, si effettua l'interazione con i contratti di Zonia e si ottengono i valori dei dati relativi ai sensori della polizza.

4. **Esecuzione automatica del contratto:** lo smart contract valuta i dati ricevuti; se le condizioni parametriche vengono soddisfatte, procede alla liquidazione automatica del sinistro.

5. **Gestione della scadenza:** qualora il periodo di validità della polizza termini senza l'attivazione di un evento assicurativo, il contratto ne segnala la scadenza, permettendo alla compagnia di riscattare la polizza e aggiornare i propri registri.

Questo flusso garantisce una gestione continua e autonoma dell'intero ciclo assicurativo, riducendo tempi di risposta, rischi operativi e costi amministrativi. Inoltre, la possibilità di monitorare lo stato delle polizze in tempo reale consente una maggiore trasparenza e un controllo più efficace sia per l'assicurato che per la compagnia.

2.6 Ruoli e interazioni tra i componenti

Il sistema coinvolge diversi attori che interagiscono in modo coordinato:

- **Utente:** sottoscrive la polizza, definisce i parametri di rischio e monitora l'andamento delle condizioni ambientali.
- **Compagnia assicurativa:** stabilisce le regole contrattuali, le soglie di attivazione e gestisce le polizze in scadenza.
- **Zonia:** agisce come ponte di fiducia tra mondo fisico e digitale, certificando i dati provenienti dai sensori.
- **Smart Contracts su Blockchain:** eseguono le clausole contrattuali e registrano in modo permanente le transazioni e gli eventi.
- **Applicazione Smart Insurance:** coordina e visualizza le interazioni tra tutti i livelli, offrendo all'utente un'interfaccia chiara e affidabile.

Questa suddivisione dei ruoli assicura un elevato grado di interoperabilità e riduce le ambiguità operative.

2.7 Sicurezza e affidabilità del sistema

La sicurezza dei dati e la fiducia nel processo sono elementi centrali dell'architettura proposta. L'integrazione di Zonia come fonte certificata riduce il rischio di falsificazioni o manomissioni dei dati. Inoltre, l'uso della blockchain garantisce che ogni operazione venga registrata in modo immutabile e consultabile.

Ogni componente dell'ecosistema contribuisce all'affidabilità complessiva:

- Zonia assicura la qualità e autenticità dei dati IoT.
- Gli smart contracts eliminano l'intervento umano, riducendo il rischio di errore o manipolazione.
- L'applicazione fornisce meccanismi di autenticazione e accesso controllato per gli utenti.

Grazie a questa combinazione, il sistema raggiunge un equilibrio tra decentralizzazione, automazione e controllo, rendendo le operazioni verificabili e resistenti ad attacchi o errori sistemici.

Capitolo 3

Implementazione

3.1 Introduzione

Il capitolo presenta l'implementazione dell'applicazione ibrida, una dApp progettata per gestire la creazione e l'esecuzione di polizze assicurative parametriche (*Smart Insurances*) basate su dati ambientali reali. L'obiettivo è fornire una piattaforma decentralizzata, trasparente e automatizzata, che unisca in un unico ecosistema le funzionalità di gestione dei contratti, l'interazione utente-compagnia e la verifica dei dati attraverso Zonia.

L'applicazione è sviluppata con React Native e opera sulla *testnet Sepolia* di Ethereum. L'interazione con la blockchain è mediata da *Ethers.js*, mentre la connessione dei wallet è garantita dal protocollo *WalletConnect v2*. Il sistema utilizza un token ERC-20 denominato *TulToken*, impiegato per la gestione di premi e rimborsi.

3.2 Architettura On-Chain

Il nucleo decentralizzato dell'applicazione è costituito da un insieme di contratti smart interconnessi. Ciascun contratto svolge un ruolo specifico e contribuisce alla logica complessiva del sistema. I principali componenti sono:

- **UserCompanyRegistry:** gestisce la registrazione e l'autenticazione dei wallet, creando un collegamento univoco tra ciascun wallet e la relativa istanza di *IndividualWalletInfo*.

- **IndividualWalletInfo:** rappresenta il profilo on-chain di un utente o di una compagnia, contenente l'elenco delle polizze sottoscritte o emesse.

- **SmartInsurance:** contratto principale per la definizione di una singola polizza assicurativa parametrica, configurata con dati ambientali, valori soglia e localizzazione geografica. Comprende sia le informazioni della singola polizza, sia quelle per l'interazione con il sistema Zonia.

- **TulToken:** token ERC-20 utilizzato come valuta interna della piattaforma. Lo scambio di premio e rimborso delle polizze avviene tramite questo token.

Ogni nuova polizza comporta il deploy di una nuova istanza del contratto *SmartInsurance*, configurata con i parametri specifici della copertura.

3.3 Ciclo di Vita della Polizza

Ogni *Smart Insurance* segue una sequenza logica di stati, che definisce l'intero ciclo di vita della copertura assicurativa:

1. **Creazione (Pending):** la compagnia emette una nuova polizza e ne definisce i parametri principali. La polizza ora compare all'utente che può decidere di attivarla pagando il premio stabilito.

2a. **Attivazione (Active):** l'utente paga il premio e la polizza diventa attiva, pronta per essere verificata. A questo punto la polizza è irreversibile.

2b. **Cancellazione (Cancelled):** la compagnia annulla la polizza prima che l'utente la possa attivare e si conclude il ciclo.

3. **Verifica Dati (Zonia Request):** tramite il sistema Zonia, viene eseguita una richiesta di dati ambientali. Se i valori raccolti superano la soglia target, il contratto passa alla fase successiva, l'utente è abilitato a richiedere il rimborso.

4a. **Erogazione Rimborso (Claimed):** il contratto trasferisce automaticamente l'importo di `payoutAmount` all'assicurato e si chiude la polizza.

4b. **Scadenza (Expired):** la polizza raggiunge la fine del proprio periodo di validità e viene reclamata dalla compagnia.

Per garantire che una volta soddisfatte le condizioni di rimborso della polizza, il trasferimento del rimborso venga effettuato, la compagnia deposita in fase di creazione l'ammontare del valore di *payoutAmount* nel contratto della smart insurance, questo valore le verrà restituito nel caso in cui la polizza si chiude senza arrivare allo stato *Claimed*, ovvero se scade o viene cancellata prima dell'attivazione.

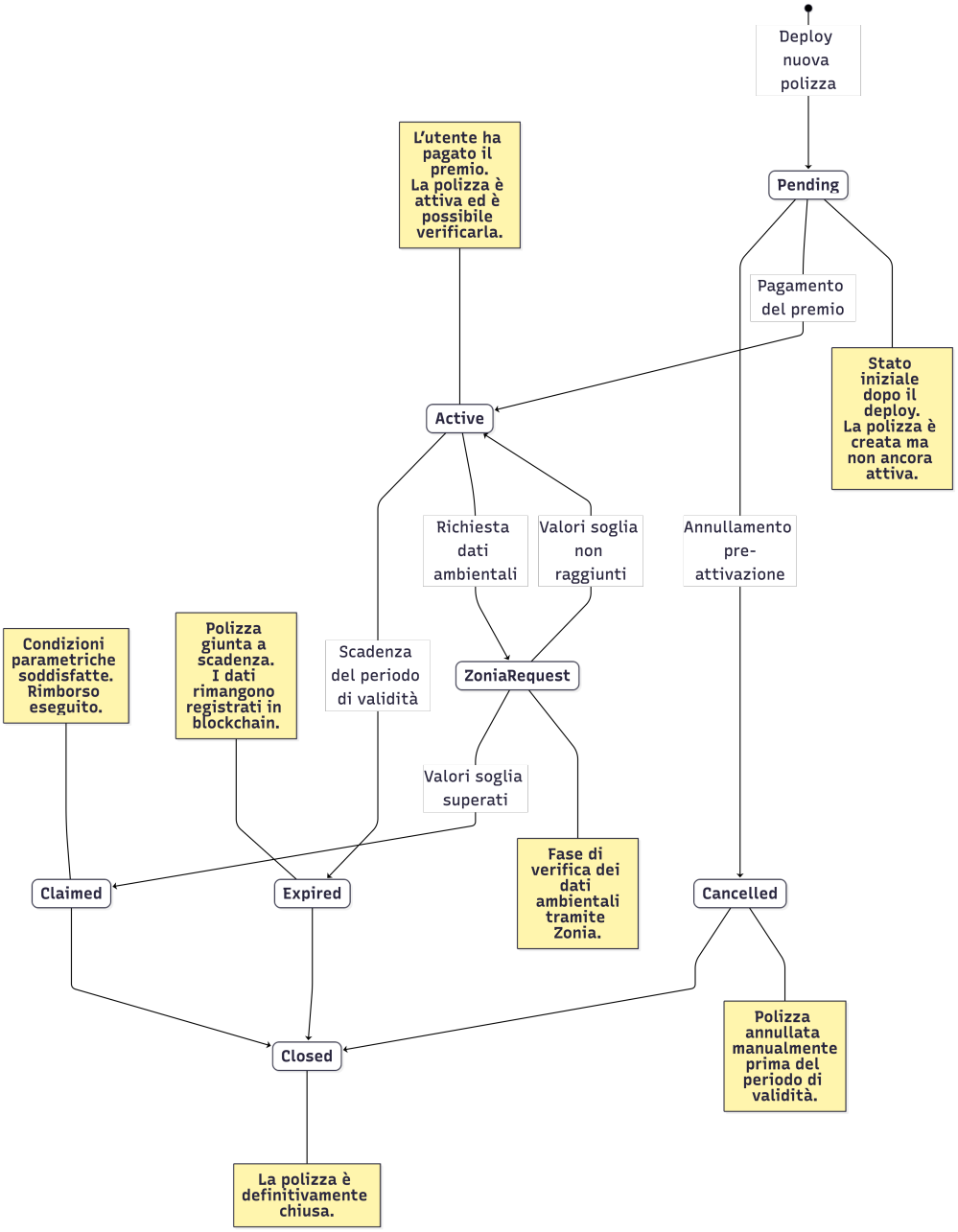

Figura 3.1: Flusso del ciclo di vita di una Smart Insurance.

3.4 Architettura Frontend

L'applicazione mobile è sviluppata con il framework React Native e si fonda su un'architettura modulare e scalabile, concepita per garantire separazione delle responsabilità, riusabilità dei componenti e semplicità di manutenzione. L'obiettivo principale del frontend è fornire un'interfaccia intuitiva e fluida che permetta all'utente di interagire con i contratti on-chain senza percepire la complessità delle tecnologie sottostanti.

3.4.1 Struttura generale e navigazione

Il frontend è organizzato in tre macro-sezioni:

1. **Autenticazione e connessione wallet:** l'utente collega il proprio wallet tramite il protocollo *WalletConnect v2*. La connessione viene mediata da un provider sicuro e sincronizzata nel contesto globale dell'app.

2. **Dashboard operativa:** visualizza le polizze attive, quelle in attesa e quelle scadute. Ogni voce è dinamicamente collegata ai contratti on-chain, consentendo aggiornamenti in tempo reale sullo stato.

3. **Gestione e visualizzazione polizze:** include modali e schermate dedicate per la creazione, l'attivazione e la consultazione delle assicurazioni, oltre alle richieste di payout automatiche.

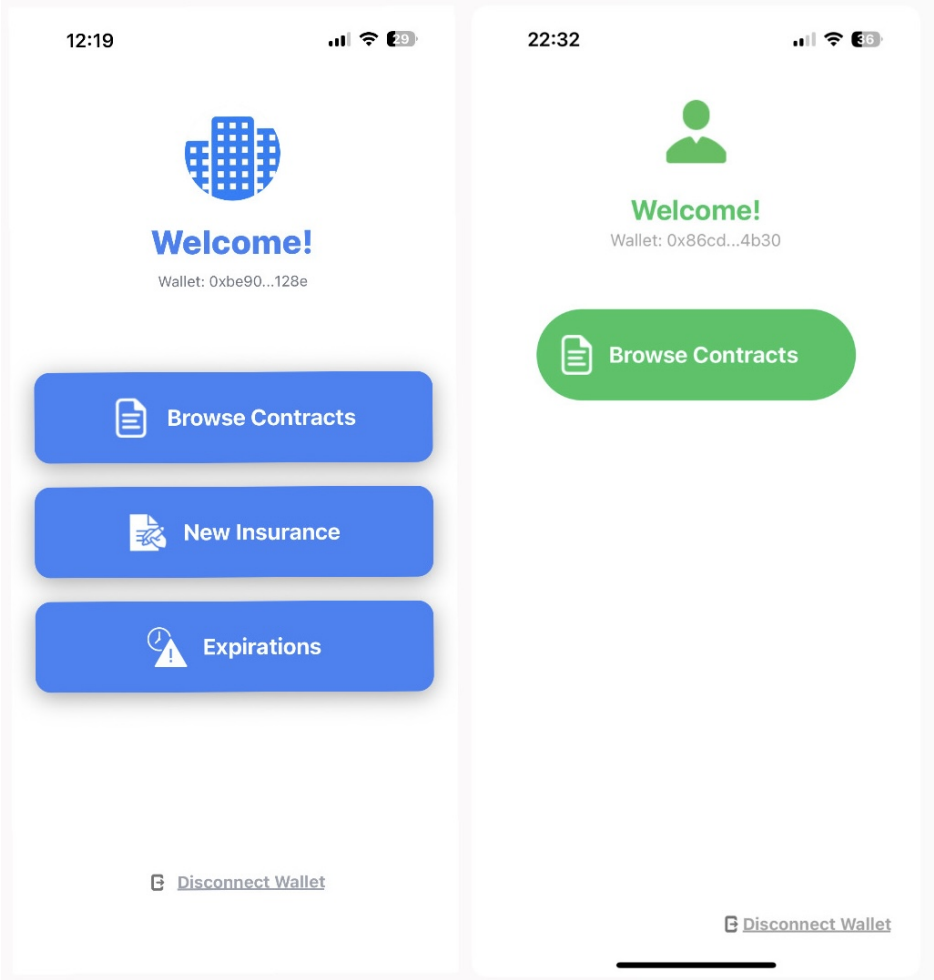

Figura 3.2: Schermata principale della dashboard di Trust App.

3.4.2 Gestione dello stato e architettura logica

Il cuore logico del frontend è costituito dal file `AuthContext.tsx`, basato sulla *Context API* di React. Questo componente funge da orchestratore dell'intero flusso applicativo e gestisce:

- la connessione e disconnessione dei wallet;
- le funzioni di interazione con i contratti (deploy, pagamento premi, verifica e payout);

- la sincronizzazione dello stato delle polizze con la blockchain in tempo reale;
- la memorizzazione del ruolo attivo dell'utente (User o Company) e la logica di navigazione condizionata.

L'interazione con la blockchain è gestita tramite *Ethers.js*, che consente di creare istanze dei contratti a partire dagli *ABI* e di effettuare transazioni firmate, riducendo il rischio di errori e semplificando il codice applicativo.

3.4.3 Componenti UI e stile

La parte visuale dell'applicazione utilizza la libreria *NativeWind*, un'estensione di *TailwindCSS* adattata a React Native. Questa scelta permette di mantenere una sintassi coerente con quella del web, accelerando lo sviluppo e garantendo uno stile uniforme. Ogni componente è progettato come elemento riutilizzabile, seguendo un approccio *component-driven development*.

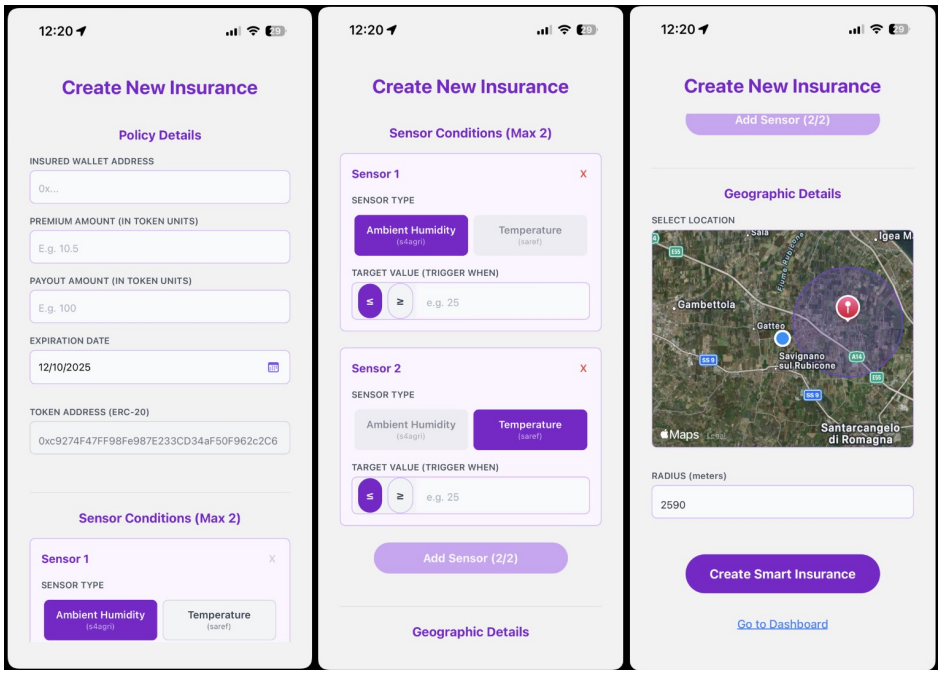

Figura 3.3: Schermata di creazione di una nuova polizza assicurativa.

3.4.4 Integrazione con la logica on-chain

Il frontend non interagisce direttamente con i sensori o con il sistema Zonia, ma agisce come ponte tra l'utente e la logica on-chain. Ogni azione dell'utente — come il pagamento di un premio o la richiesta di verifica dei dati — corrisponde a una chiamata a funzioni specifiche del contratto *SmartInsurance*. Questo approccio garantisce trasparenza e coerenza con il ciclo di vita della polizza, minimizzando le possibilità di errore e migliorando la tracciabilità.

3.5 Esperienza Utente e Flusso Operativo

L'esperienza utente (*User Experience, UX*) dell'applicazione è progettata per combinare semplicità di utilizzo e rigore tecnico, permettendo a utenti non esperti di interagire con la blockchain senza doversi confrontare direttamente con la complessità dei processi sottostanti. Ogni azione dell'utente, dall'attivazione della polizza alla richiesta di rimborso, corrisponde a una (o più) transazione reale sulla rete Ethereum, firmata digitalmente tramite WalletConnect e tracciabile pubblicamente.

Il flusso operativo è stato strutturato secondo un approccio sequenziale ma modulare, che rispecchia le fasi di vita di una polizza parametrica e ne garantisce la coerenza on-chain. Le cinque fasi principali sono le seguenti:

1. **Connessione e autenticazione:** l'utente avvia l'applicazione e collega il proprio wallet Ethereum tramite *WalletConnect v2*. In caso di primo accesso, viene guidato nella selezione del proprio ruolo operativo — *User* (assicurato) o *Company* (compagnia assicurativa) — che determinerà i permessi e le funzioni disponibili nell'interfaccia.

2. **Creazione della polizza:** la compagnia assicurativa definisce una nuova *Smart Insurance*, specificando parametri come soglia ambientale, coordinate geografiche, valore del premio e ammontare del payout. A seguito della compilazione del modulo, viene eseguita una transazione

di *deploy* del contratto sulla blockchain di test *Sepolia*, generando un indirizzo univoco associato alla polizza.

3. **Attivazione da parte dell'utente:** l'assicurato seleziona una delle polizze disponibili, ne visualizza i dettagli e, tramite l'apposita modale, effettua il pagamento del premio. La transazione di attivazione cambia lo stato del contratto da *Pending* a *Active*, rendendo la copertura effettiva.

4. **Verifica dei dati ambientali:** durante la validità della polizza, l'utente o la compagnia possono richiedere la verifica dei dati ambientali. La richiesta viene inviata a **Zonia**, che restituisce i valori autenticati. Se la soglia predefinita viene superata, il contratto aggiorna automaticamente il proprio stato.

5. **Erogazione del payout:** in caso di esito positivo, l'utente può richiedere l'esecuzione della clausola assicurativa. Il contratto effettua il trasferimento del `payoutAmount` al wallet dell'assicurato in modo automatico, completando così il ciclo di vita della polizza. Tutte le operazioni rimangono tracciate on-chain, garantendo trasparenza e immutabilità.

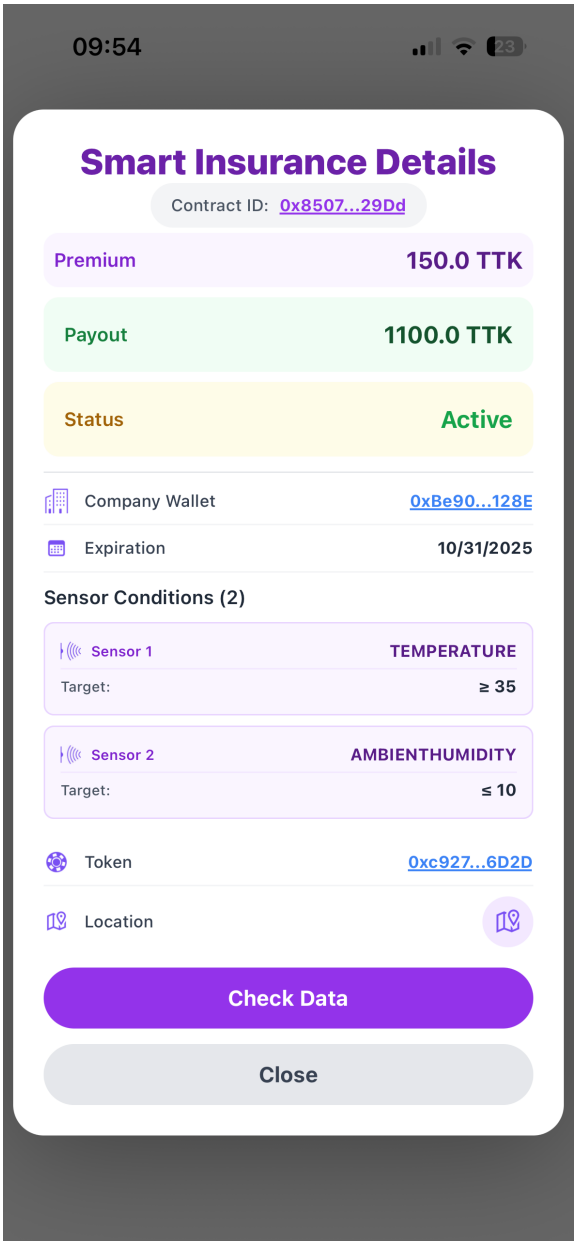

Figura 3.4: Schermata di dettaglio di una Smart Insurance.

Dal punto di vista progettuale, il flusso operativo si distingue per la sua coerenza con la struttura on-chain dei contratti: ogni azione dell'utente corrisponde a uno stato preciso del contratto *SmartInsurance*. Questo allineamento diretto tra interfaccia e logica blockchain riduce drasticamente la

possibilità di incongruenze, migliorando l'affidabilità del sistema.

Inoltre, l'applicazione introduce un approccio *user-centric* alla gestione dei contratti decentralizzati: l'utente non deve conoscere i dettagli tecnici di una transazione, ma può agire tramite un'interfaccia grafica semplificata che esegue automaticamente le chiamate necessarie attraverso *Ethers.js*. L'esperienza complessiva risulta così intuitiva, ma al tempo stesso pienamente conforme ai principi di trasparenza e decentralizzazione propri della tecnologia blockchain.

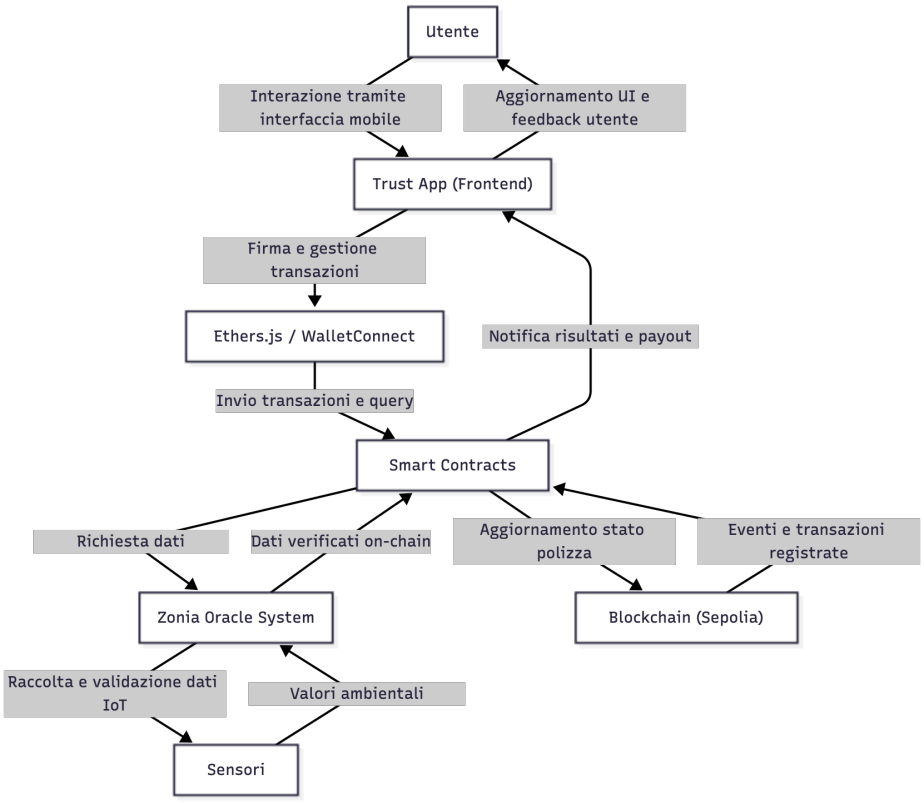

Figura 3.5: Diagramma del flusso operativo.

3.6 Conclusione

L'implementazione di **Trust App** dimostra la fattibilità di un sistema assicurativo decentralizzato in grado di coniugare sicurezza, affidabilità e

trasparenza. Il valore distintivo dell'applicazione risiede nella sinergia tra l'automazione degli *smart contracts* e la validazione decentralizzata dei dati ambientali fornita dal sistema **Zonia**, aprendo la strada a nuove forme di assicurazioni parametriche basate su informazioni reali e certificate, pur mantenendo un'esperienza d'uso semplice e accessibile. Le soluzioni realizzate costituiscono un modello solido su cui basare ulteriori evoluzioni del sistema.

Il capitolo successivo è dedicato alla fase di **testing**, focalizzata sulla valutazione delle funzionalità principali dell'applicazione e sull'analisi dei costi di esecuzione (*gas fees*) delle transazioni on-chain, con l'obiettivo di verificare la sostenibilità ed efficienza complessiva del modello proposto.

Capitolo 4

Testing

4.1 Introduzione

La fase di testing rappresenta un passaggio cruciale per la validazione del sistema proposto, poiché consente di verificare la correttezza funzionale delle componenti e di valutare l'efficienza economica delle transazioni eseguite sulla blockchain. Nel contesto dell'applicazione, le attività di test hanno riguardato principalmente due aspetti:

1. la verifica del corretto funzionamento degli smart contracts e dei flussi operativi che regolano il ciclo di vita delle polizze;

2. l'analisi dei costi di esecuzione (*gas fees*) associati alle principali operazioni on-chain.

Gli obiettivi del testing sono stati dunque duplici: da un lato garantire la stabilità logica del sistema e l'affidabilità dei processi automatizzati, dall'altro valutare la sostenibilità del modello in un ambiente blockchain reale, prendendo come riferimento la *testnet Sepolia* di Ethereum.

4.2 Ambiente di Test

Le prove sono state condotte in un ambiente controllato, replicando le condizioni di utilizzo reali dell'applicazione. La testnet *Sepolia* è stata scelta come rete di riferimento per la compatibilità con i protocolli Ethereum e la disponibilità di faucet pubblici per la gestione dei token di test. L'applicazione è stata configurata per interagire con la blockchain tramite il provider di rete di *Ethers.js*, mentre la gestione dei wallet è avvenuta attraverso il protocollo *WalletConnect v2*.

Un ruolo centrale nelle attività di verifica è stato svolto da **Etherscan**, la piattaforma di esplorazione dei blocchi di Ethereum, utilizzata per monitorare in tempo reale la creazione dei contratti, la propagazione delle transazioni e l'aggiornamento degli stati on-chain. Tramite Etherscan è stato possibile validare la corretta esecuzione dei metodi, analizzare il consumo di *gas* per ciascuna operazione e confermare la coerenza tra i risultati visualizzati nell'applicazione e quelli registrati effettivamente sulla rete.

L'utilizzo combinato di strumenti di sviluppo locali e di servizi di monitoraggio esterni ha permesso di ottenere una visione completa del comportamento del sistema, consentendo di identificare eventuali anomalie o ritardi nella propagazione delle transazioni in fase di test. In particolare, Etherscan si è rivelato uno strumento essenziale per l'analisi post-esecuzione, permettendo di verificare in modo indipendente la corretta finalizzazione dei blocchi e la persistenza degli eventi registrati on-chain.

Le principali componenti coinvolte nei test sono:

- **Smart Contracts:** distribuiti e testati tramite *Hardhat* e *Ethers.js*, includendo le funzioni di creazione, attivazione e chiusura delle polizze;
- **Applicazione mobile:** eseguita in ambiente *Expo* su dispositivi Android e iOS, con connessione wallet gestita tramite *WalletConnect v2*;
- **Zonia Oracle System:** eseguito sempre sulla testnet, seguendo il deploy ufficiale del contratto.

4.3 Verifica Funzionale dei Contratti

La prima fase del testing è stata dedicata alla verifica funzionale dei contratti intelligenti, al fine di confermare che il ciclo di vita delle polizze si svolgesse in maniera coerente con il modello progettuale.

Le prove hanno riguardato in particolare:

- la **creazione** di nuove polizze da parte delle compagnie e la corretta inizializzazione dei parametri ambientali;
- l'**attivazione** da parte dell'utente con versamento del premio;
- la **verifica automatica** dei dati ambientali forniti da Zonia e la conseguente esecuzione del payout;
- la gestione delle **scadenze temporali** e la chiusura automatica delle polizze non attivate.

Durante le prove, è stato verificato che ogni evento generasse una o più transazioni tracciabili in blockchain e che gli stati della polizza si aggiornassero correttamente, senza possibilità di conflitti o duplicazioni. L'uso di variabili temporali (`timestamp`) e meccanismi di controllo delle condizioni (`require`) ha garantito la solidità logica dei contratti e la corretta esecuzione in tutte le situazioni previste.

Sono stati tuttavia riscontrati, in modo sporadico, alcuni casi isolati di **mancata ricezione della ricevuta di transazione** da parte del provider, nonostante l'operazione risultasse correttamente eseguita sulla rete *Sepolia*, come verificato tramite *Etherscan*. Tali episodi, attribuibili principalmente a momentanee instabilità della testnet o a ritardi nella propagazione dei blocchi, hanno comportato un timeout lato applicazione e la necessità di ripetere manualmente la transazione. Queste situazioni non hanno influito sulla coerenza dei dati on-chain, ma hanno evidenziato l'importanza di implementare, in contesti produttivi, meccanismi di *retry* o di conferma asincrona per la gestione robusta delle risposte dai provider blockchain.

4.4 Analisi delle Gas Fee

La sostenibilità economica del sistema è fortemente influenzata dai costi di esecuzione delle transazioni on-chain, misurati in *gas*. Per valutare le prestazioni del modello, è stata condotta un'analisi dettagliata utilizzando il plugin `hardhat-gas-reporter`, con lo scopo di stimare i consumi medi associati alle principali funzioni degli smart contract e di tradurli in valori economici realistici.

4.4.1 Metodologia di test

I test sono stati eseguiti localmente in ambiente Hardhat, replicando il comportamento della rete Ethereum e utilizzando un **gas price** fisso pari a **3 GWEI**. La conversione da ETH a USD è stata automatizzata tramite l'integrazione dell'API di CoinMarketCap, che al momento del test riportava un valore medio di **1 ETH = 4126,76 USD**.

Il compilatore Solidity è impiegato con l'ottimizzatore disattivato e 200 cicli di esecuzione (*runs*), per garantire la riproducibilità dei risultati e misurare i consumi in condizioni standard.

4.4.2 Risultati sperimentali

La Tabella 4.1 riassume i risultati ottenuti per le principali operazioni del sistema, considerando sia le funzioni più ricorrenti sia le fasi chiave del ciclo di vita di una polizza.

Operazione	Gas Medio	Costo (USD)
Deploy SmartInsurance	4 039 353	50,01
Deposito payout polizza	165 746	2,05
Pagamento premio	170 416	2,11
Invio richiesta Zonia	932 885	11,54
Verifica dati Zonia	68 344	0,85
Esecuzione del payout	167 448	2,07
Cancellazione polizza	161 286	2,00
Aggiornamento scadenza	158 364	1,96

Tabella 4.1: Consumi medi di gas e costi in USD (3 GWEI, ETH = 4126,76 USD) misurati con Hardhat Gas Reporter.

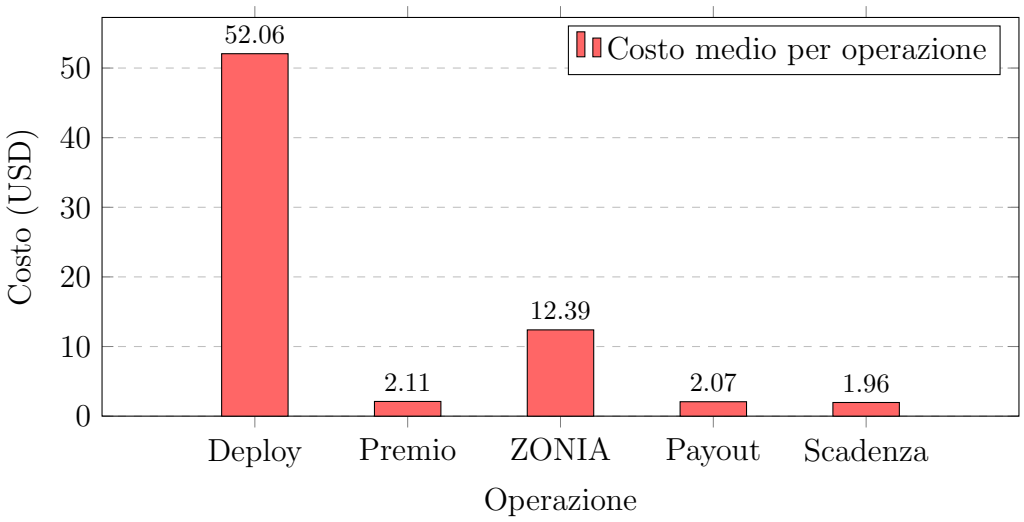

Figura 4.1: Confronto tra i costi delle principali operazioni on-chain.

4.4.3 Discussione dei risultati

Dall'analisi emerge che il **deploy iniziale delle smart insurance** è l'operazione più dispendiosa in termini di gas, ma viene eseguita una sola volta per ogni assicurazione. Le operazioni successive — tra cui il pagamento del

premio, la verifica tramite Zonia e il payout — hanno un impatto economico trascurabile.

Infine, i valori raccolti suggeriscono che l'adozione di un meccanismo di **commissioni dinamiche** — proporzionali alla complessità e al costo reale delle operazioni — potrebbe costituire una soluzione equilibrata per garantire la sostenibilità del servizio anche in contesti di rete più congestionata, mantenendo la convenienza economica per gli utenti finali.

4.5 Testing dell'Applicazione

Parallelamente alle verifiche on-chain, è stata condotta un'estesa attività di testing dell'interfaccia utente e delle logiche applicative di *Trust App*. Questa fase aveva l'obiettivo di assicurare che l'applicazione fungesse da ponte affidabile tra la componente blockchain e l'esperienza dell'utente finale, validando così la solidità complessiva dell'ecosistema.

Gli obiettivi principali sono stati:

- garantire la stabilità della connessione *WalletConnect* durante le operazioni di firma e invio delle transazioni;
- validare la corretta sincronizzazione tra lo stato on-chain dei contratti e le informazioni mostrate nel frontend;
- verificare la coerenza dei dati visualizzati (polizze attive, rimborsi ricevuti, scadenze) in tutte le sessioni di utilizzo;
- assicurare un'esperienza di navigazione fluida anche in condizioni di rete instabili o in caso di errori temporanei del provider.

Le prove sono state eseguite su emulatori mobili e dispositivi reali, utilizzando l'ambiente *Expo Go* e strumenti di logging interni per la raccolta e l'analisi dei comportamenti. Durante i test, è stato verificato che l'applicazione reagisse correttamente a ogni evento proveniente dalla blockchain o dal

sistema Zonia, aggiornando in tempo reale l'interfaccia utente senza ritardi o desincronizzazioni.

L'attività di testing ha permesso di validare non solo la stabilità tecnica del sistema, ma anche la sua **usabilità**, confermando che la complessità della gestione dei contratti e delle interazioni blockchain può essere nascosta dietro un'interfaccia semplice e intuitiva. Questa evidenza è cruciale nel contesto della tesi, poiché dimostra che l'adozione di tecnologie decentralizzate nel settore assicurativo è possibile senza compromettere la semplicità d'uso per l'utente finale.

4.6 Discussione dei Risultati e Conclusioni

L'analisi complessiva dei test ha confermato la robustezza, la coerenza funzionale e l'affidabilità dell'architettura proposta. Gli smart contracts hanno eseguito correttamente tutte le clausole previste, gestendo in modo coerente l'intero ciclo di vita delle polizze, dalla creazione alla chiusura. L'interfaccia utente, pur mantenendo un approccio minimalista, si è dimostrata chiara e funzionale, garantendo un'interazione efficace sia per gli utenti sia per le compagnie assicurative.

Dal punto di vista economico, l'analisi delle *gas fee* ha evidenziato come il sistema sia sostenibile anche in scenari di utilizzo intensivo. L'operazione di **deploy della polizza** rappresenta il costo principale, ma viene eseguita una sola volta per ciascun contratto; le operazioni successive (attivazione, verifica tramite ZONIA, chiusura) comportano costi contenuti, confermando la praticabilità del modello anche in un contesto reale.

Nel complesso, *Trust App* ha dimostrato di poter supportare un modello di assicurazione parametrica completamente automatizzato, trasparente e decentralizzato, in cui la fiducia non è più riposta in un ente centrale ma nei meccanismi stessi della blockchain e degli oracoli certificati. Il sistema combina affidabilità tecnica e sostenibilità economica, ponendosi come base per future evoluzioni nel settore delle smart insurance.

I risultati ottenuti aprono inoltre la strada a riflessioni sulla gestione delle **commissioni dinamiche**, un aspetto che potrebbe rafforzare ulteriormente la sostenibilità del modello. La definizione di strategie di calcolo adattive, basate sul carico della rete o sulla complessità delle polizze, rappresenta una naturale estensione del lavoro presentato. Questa prospettiva sarà approfondita nel capitolo successivo, dedicato all'analisi dei costi di rete e all'ottimizzazione economica del sistema assicurativo decentralizzato.

Capitolo 5

Conclusioni e Sviluppi Futuri

5.1 Discussione Finale

Il lavoro presentato ha dimostrato la piena fattibilità tecnica e concettuale di un ecosistema assicurativo decentralizzato basato su *smart contracts*, dati certificati e un'infrastruttura di validazione esterna come **Zonia**. L'applicazione *Trust App* ha mostrato come la sinergia tra blockchain, IoT e oracoli possa trasformare radicalmente i processi assicurativi, rendendoli più efficienti, trasparenti e verificabili.

Dal punto di vista tecnico, i test hanno confermato la stabilità dell'architettura e la corretta esecuzione del ciclo di vita delle polizze, mentre l'analisi delle *gas fee* ha fornito indicazioni concrete sulla sostenibilità economica del modello. Pur con alcune limitazioni legate all'utilizzo della rete di test, l'applicazione ha dimostrato la capacità di gestire in modo affidabile la creazione, l'attivazione e la chiusura automatica delle polizze, mantenendo coerenza tra gli stati on-chain e l'interfaccia utente. Nel complesso, il sistema proposto rappresenta un esempio tangibile di come la tecnologia decentralizzata possa essere applicata con successo a un settore tradizionalmente complesso come quello assicurativo.

5.2 Contributo Innovativo del Modello Proposto

Rispetto alle soluzioni esistenti nel panorama delle *smart insurance*, il modello sviluppato introduce elementi di innovazione sostanziale, che ne costituiscono il principale valore aggiunto e ne rafforzano la rilevanza scientifica e applicativa:

1. **Integrazione nativa con un sistema di validazione certificato:** a differenza dei modelli tradizionali che si affidano a oracoli generici o fonti di dati non verificate, il sistema proposto utilizza esclusivamente dati ambientali autenticati e tracciabili forniti da **Zonia**. Questo garantisce integrità e affidabilità delle informazioni, riducendo in modo significativo il rischio di discrepanze o manipolazioni.

2. **Gestione completa e autonoma del ciclo di vita della polizza:** ogni smart contract include meccanismi di scadenza e chiusura automatica, offrendo una gestione realmente end-to-end del portafoglio assicurativo. Ciò consente una riduzione dei costi gestionali e una maggiore trasparenza per tutti gli attori coinvolti.

3. **Esperienza utente semplificata e accesso trasversale:** grazie a un'interfaccia mobile sviluppata con React Native, l'utente può interagire con un sistema basato su blockchain senza percepirne la complessità, mantenendo una piena trasparenza sulle operazioni on-chain.

In sintesi, il modello proposto supera la visione puramente automatizzata delle smart insurance, proponendo invece un sistema che coniuga **affidabilità dei dati, automazione contrattuale e sostenibilità economica**, creando così le basi per una nuova generazione di soluzioni assicurative decentralizzate.

5.3 Proposta di Commissioni Dinamiche e Linee Guida Economiche

Uno degli aspetti più rilevanti emersi nella fase di testing riguarda la necessità di un meccanismo di **commissioni dinamiche**, volto a garantire la sostenibilità economica del sistema e la remunerazione del gestore dell'infrastruttura.

Ogni operazione on-chain comporta un costo di transazione (*gas fee*), ma la sola copertura di tali costi non assicura la sopravvivenza economica dell'ecosistema. Si propone pertanto un modello ibrido di **commissioni di servizio** che coniughi equità per l'utente e redditività per il gestore, fondato su tre principi cardine:

1. **Equità:** le commissioni devono essere proporzionali al tipo e alla complessità dell'operazione, mantenendo accessibile l'utilizzo del sistema.

2. **Sostenibilità:** il gestore deve poter coprire i costi infrastrutturali (monitoraggio, validazione, manutenzione) e generare un margine di profitto stabile e trasparente.

3. **Adattività:** le commissioni devono adattarsi in modo dinamico alle condizioni della rete, al volume delle polizze attive e al numero di interazioni con gli oracoli.

5.3.1 Struttura proposta delle commissioni

La struttura economica proposta si articola in due componenti:

- una **commissione base fissa** (C_{base}), destinata al gestore del sistema per coprire i costi operativi minimi;

- una **commissione variabile dinamica**, calcolata in funzione del gas medio della rete, del numero di polizze attive e della frequenza delle richieste a Zonia.

La relazione complessiva può essere espressa come:

$$C_{tot} = C_{gas} + C_{base} + \alpha \cdot f(C_{rete}, N_{polizze}, R_{Zonia})$$

dove:

- C_{gas} rappresenta il costo reale della transazione on-chain;
- C_{base} è la quota fissa di servizio (ad esempio tra 0.5 e 1 USD);
- α è un coefficiente di bilanciamento impostato dal gestore;
- $f(\cdot)$ è una funzione adattiva che modula le commissioni in base alla congestione della rete, al numero di polizze attive e al volume delle richieste oracolari.

Questo modello consente di ottenere un flusso di ricavi proporzionale all'utilizzo effettivo del sistema, mantenendo allo stesso tempo un'esperienza d'uso stabile e prevedibile. In condizioni di elevata congestione di rete, le commissioni aumentano in modo controllato per garantire la sostenibilità, mentre in periodi di bassa attività si riducono, incentivando l'uso continuativo della piattaforma.

5.3.2 Equilibrio tra profitto e accessibilità

L'obiettivo economico del modello non è la massimizzazione del profitto, ma la creazione di un ecosistema sostenibile. Un sistema efficiente deve trovare un equilibrio tra la redditività del gestore e l'accessibilità per gli utenti, evitando che le commissioni rappresentino una barriera d'ingresso. La logica di **commissioni dinamiche** proposta favorisce questo equilibrio, adattandosi al contesto operativo e garantendo la continuità economica del servizio nel tempo.

5.4 Sviluppi Futuri

Il lavoro apre a diverse prospettive di ricerca e sviluppo, che potrebbero consolidare ulteriormente il modello proposto:

- **Formalizzazione matematica del modello di commissioni dinamiche:** sviluppo di formule predittive basate su modelli di apprendimento automatico per anticipare variazioni del costo del gas e ottimizzare il calcolo delle fee.

- **Automazione della gestione economica:** implementazione di smart contracts dedicati alla distribuzione automatica delle commissioni tra gestore, compagnie e utenti, garantendo piena trasparenza e tracciabilità.

- **Migrazione verso soluzioni Layer 2:** adozione di reti come *Arbitrum*, *Optimism* o *Base* per ridurre drasticamente i costi di transazione e migliorare la scalabilità [13].

- **Estensione multi-asset:** introduzione del supporto a token multipli o stablecoin per aumentare la flessibilità dei pagamenti e ridurre l'esposizione alla volatilità dell'ETH.

- **Dashboard analitica per il gestore:** sviluppo di una piattaforma di monitoraggio che permetta di analizzare in tempo reale le metriche di rete, le transazioni e i profitti generati dal sistema.

Queste linee di sviluppo rappresentano la naturale evoluzione di un sistema che, da prototipo accademico, può maturare fino a diventare una piattaforma pienamente operativa nel mercato delle assicurazioni decentralizzate.

5.5 Conclusione

La tesi ha dimostrato la validità di un modello assicurativo basato su smart contracts e dati ambientali certificati, capace di unire automazione, trasparenza e affidabilità. L'introduzione del concetto di **commissioni dinamiche** e la definizione di linee guida economiche concrete rappresentano un passo decisivo verso la sostenibilità di lungo periodo del sistema.

Trust App non è soltanto un prototipo tecnologico, ma un modello operativo che coniuga innovazione e realismo, dimostrando che un approccio decentralizzato può essere al tempo stesso sicuro, accessibile e economicamente sostenibile. In prospettiva, questo lavoro rappresenta una base solida per lo sviluppo di un nuovo paradigma di **smart insurance**, in cui fiducia, automazione e sostenibilità convivono in un equilibrio capace di ridefinire il futuro del settore assicurativo.

Bibliografia

[1] K. Christidis and M. Devetsikiotis, "Blockchains and smart contracts for the internet of things," *IEEE Access*, vol. 4, pp. 2292–2303, 2016.

[2] T.-L. Teh and C. Woolnough, "A better trigger: Indices for insurance," *Journal of Risk and Insurance*, vol. 86, no. 4, pp. 861–885, 2018.

[3] A. Marr, A. Winkel, M. van Asseldonk, R. Lensink, and E. Bulte, "Adoption and impact of index-insurance and credit for smallholder farmers in developing countries: A systematic review," *Agricultural Finance Review*, vol. 76, no. 1, pp. 94–118, 2016.

[4] N. D. Jensen, A. G. Mude, and C. B. Barrett, "The welfare effects of index-based livestock insurance: Livestock herding on communal lands," *Environmental and Resource Economics*, vol. 78, pp. 587–613, 2021.

[5] G. Elabed, M. Bellemare, M. Carter, and C. Guirkinger, "Managing basis risk with multiscale index insurance," *Agricultural Economics*, vol. 44, 07 2013.

[6] M. Stigler and D. Lobell, "On the benefits of index insurance in us agriculture: a large-scale analysis using satellite data," *arXiv preprint arXiv:2011.12544*, 2020.

[7] S. Surminski and A. H. Thieken, "Parametric insurance for climate risk: applications in urban environments," *Climate Risk Management*, vol. 32, p. 100283, 2021.

[8] R. G. Subbian and P. K. Gollapudi, "Smart contracts and blockchain - the next frontier in trustworthy p&c insurance," *International Journal of Computer & Organization Trends*, vol. 15, no. 1, pp. 41–53, 2025.

[9] M. Wang and H. Assa, "Blockchain and contract theory: modeling smart contracts using insurance markets," *Managerial Finance*, vol. 44, no. 4, pp. 478–494, 2018.

[10] L. Gao, W. Li, and D. Xu, "Data provenance and integrity in iot-based smart insurance," *Future Generation Computer Systems*, vol. 110, pp. 613–624, 2020.

[11] L. Gigli, I. Zyrianoff, F. Montori, L. Sciullo, C. Kamienski, and M. D. Felice, "Zonia: A zero-trust oracle system for blockchain iot applications," *IEEE Internet of Things Journal*, pp. 1–1, 2025.

[12] A. Đurović, "Smart contracts as an innovation in insurance law," *Pravo i privreda*, no. 3, pp. 305–317, 2020.

[13] A. Gangwal, H. R. Gangavalli, and A. Thirupathi, "A survey of layer-two blockchain protocols," *arXiv preprint arXiv:2204.08032*, vol. 2204.08032, 2022. version 3, revised 26 Jul 2022.

Ringraziamenti

Alla mia famiglia,
perché il desiderio di rendervi orgogliosi
ha sempre prevalso su qualsiasi difficoltà.

A mia mamma,
che mi ha trasmesso la pazienza e la perseveranza.

A mio babbo,
che mi ha insegnato il valore del sacrificio.

A mio fratello Luca,
per il suo sostegno silenzioso ma costante.

A Ilaria,
mio porto sicuro negli attimi di incertezza,
fonte di gioia per i traguardi raggiunti
e mio conforto nei momenti difficili.
Al tuo "Sei bravissimo, ce la farai" — che mi ha sempre dato la forza.

Ai miei amici,
che hanno saputo esserci con leggerezza, ironia e affetto.

A tutti voi,
che con pazienza e fiducia mi avete accompagnato in ogni passo:
questo traguardo è anche vostro.