POLITECNICO DI TORINO

TESINA BLOCKCHAIN E CRIPTOECONOMIA

CHAINLINK

Attanasi Alessio
Buccelli Giorgia
Cannistrà Maria
El Amrani Rachid

ANNO ACCADEMICO 2022/2023

Indice

1 Introduzione 3

2 Architettura 5
- 2.1 Architettura on-chain . . . 6
- 2.2 Architettura off-chain . . . 8

3 Sicurezza dell'oracolo 11
- 3.1 Oracolo ideale . . . 11

4 Approccio alla decentralizzazione di ChainLink 15
- 4.1 Fonti distributive . . . 16
- 4.2 Oracoli distributivi . . . 16

5 Servizi di Sicurezza ChainLink 19
- 5.1 Sistema di Validazione . . . 20
- 5.2 Sistema di Reputazione . . . 20
- 5.3 Servizio di certificazione . . . 21
- 5.4 Servizio di aggiornamento dei contratti . . . 21
- 5.5 Utilizzo del token LINK . . . 22

6 Off-Chain Reporting 23

7 Conclusione 25

Capitolo 1

Introduzione

Gli smart contracts sono applicazioni che vengono eseguite su un'infrastruttura decentralizzata, come una blockchain. Essi sono a prova di manomissione, ovvero nessuna parte (nemmeno il loro creatore) può alterare il loro codice o interferire con la loro esecuzione. Storicamente, i contratti incorporati nel codice venivano eseguiti in modo centralizzato, di conseguenza essi potevano essere soggetti ad alterazione, risoluzione e persino cancellazione da parte di una parte privilegiata (l'autorità centrale). Al contrario, le garanzie di esecuzione degli smart contracts, che vincolano tutte le parti a un accordo come scritto originariamente, creano un nuovo e potente tipo di relazione fiduciaria che non si basa sulla fiducia in nessuna delle parti coinvolte, bensì la fiducia è riposta nella tecnologia stessa. Poiché sono autoverificanti e autoeseguibili (ovvero, a prova di manomissione come spiegato sopra), gli smart contracts offrono quindi un veicolo superiore per realizzare e amministrare accordi digitali. Il nuovo potente modello di fiducia rappresentato dagli smart contracts, tuttavia, introduce una nuova sfida tecnica: la connettività.

La stragrande maggioranza delle interessanti applicazioni degli smart contracts si basa su dati sul mondo reale che provengono da risorse chiave, in particolare feed di dati e API, che sono esterne alla blockchain. A causa dei meccanismi di consenso alla base delle blockchain, una blockchain non può recuperare direttamente tali dati critici.

Viene proposta una soluzione al problema della connettività dello smart contract sotto forma di ChainLink, una rete di oracoli decentralizzata creata da Sergey Nazarov e Steve Ellis ed introdotta nel 2017, quando è stato presentato il whitepaper, ed è stata lanciata ufficialmente nel 2019. Chainlink fornisce transazioni sicure utilizzando fonti di dati e API esterne, consentendo a chiunque di unirsi alla rete e di fornire dati o di completare i "lavori" di Chainlink, come la gestione di nodi e oracoli affiliati.
Nel 2018, Chainlink ha integrato Town Crier, un oracolo blockchain basato su un ambiente di esecuzione affidabile che è stato co-sviluppato da Ari Juels della Cornell University. Questa integrazione ha permesso a Chainlink di collegare la blockchain di Ethereum con fonti web che utilizzano HTTPS.
Nel 2019 Chainlink ha lanciato ufficialmente il proprio protocollo, seguito dalla registrazione del marchio Chainlink nelle Isole Cayman. Nel 2020 Chainlink ha integrato DECO, un altro progetto co-creato da Juels. DECO è un protocollo che utilizza prove a conoscenza zero (zero-knowledge proofs) per consentire agli utenti di dimostrare la veridicità delle informazioni a un oracolo blockchain senza rivelare informazioni sensibili.
Chainlink ha pubblicato nel 2021 un secondo whitepaper che introduce Chainlink 2.0. Il documento espande le capacità delle reti oracolo decentralizzate e introduce smart contract ibridi che utilizzano il codice on-chain e i servizi off-chain forniti dalle reti oracolo.

Ciò che differenzia ChainLink da altre soluzioni Oracle è la sua capacità di operare come una rete completamente decentralizzata. Questo approccio decentralizzato limita la fiducia in ogni

singola parte, consentendo di estendere la qualità a prova di manomissione apprezzata negli smart contracts all'operazione end-to-end tra gli smart contracts e le API su cui si basano.
Rendere gli smart contracts capaci di interagire con le risorse off-chain è necessario se si intende sostituire gli accordi digitali oggi in uso. Esempi di potenziali smart contracts di nuova generazione e relativi requisiti di dati includono:

- Gli smart contracts su titoli come obbligazioni, derivati su tassi di interesse e molti altri richiederanno l'accesso alle API che riportano i prezzi di mercato e i dati di riferimento del mercato, ad es. tassi di interesse.
- Gli smart contracts assicurativi avranno bisogno di feed di dati sui dati IoT relativi all'evento assicurabile in questione, ad esempio: la porta magnetica del magazzino era chiusa al momento della violazione, il firewall dell'azienda era online o il volo per cui avevi l'assicurazione è arrivato in tempo.
- Gli smart contracts di trade finance avranno bisogno di dati GPS sulle spedizioni, dati dai sistemi ERP della catena di approvvigionamento e dati doganali sulle merci spedite per confermare l'adempimento degli obblighi contrattuali.

Un altro problema comune a questi esempi è l'impossibilità per gli smart contracts di inviare dati a sistemi off-chain. Tale output assume spesso la forma di un messaggio di pagamento indirizzato all'infrastruttura centralizzata tradizionale in cui gli utenti hanno già un account, ad esempio per pagamenti bancari, PayPal e altri circuiti di pagamento. La capacità di ChainLink di inviare in modo sicuro i dati alle API e a vari sistemi legacy per conto di uno smart contract consente la creazione di contratti a prova di manomissione esterni.

Dopo questa breve introduzione, verrà esaminata l'architettura della rete, presentando sia un semplice sistema di aggregazione dei dati dei contratti on-chain, sia un meccanismo di consenso off-chain più efficiente. Viene descritto anche il supporto dei servizi di monitoraggio della reputazione e della sicurezza per ChainLink che aiutano gli utenti ad effettuare selezioni informate del provider e ottenere un servizio solido. Viene illustrato, in seguito, l'approccio di ChainLink alla decentralizzazione, distribuzione e sicurezza degli oracoli, con una discussione dei quattro servizi di sicurezza proposti da ChainLink, nonché del ruolo svolto dai token LINK. Concludendo, viene presentato il protocollo di off-chain reporting, che comporta una maggiore decentralizzazione e scalabilità della rete Chainlink.

Capitolo 2

Architettura

L'architettura di Chainlink è suddivisa in due componenti che fanno da collegamento tra il mondo interno della blockchain e il mondo esterno da cui ricevere le informazioni:

- architettura on-chain, la parte della piattaforma che è eseguita direttamente sulla blockchain;
- architettura off-chain, costituita da nodi esterni alla blockchain.

La parte on-chain del sistema riguarda quegli smart contracts relativi alla richiesta di dati esterni alla blockchain e l'utilizzo di tali dati per automatizzare gli accordi e le transazioni. La parte off-chain del sistema è costituita da nodi oracolari, responsabili della raccolta, della verifica e della trasmissione di dati provenienti da diverse fonti agli smart contracts su blockchain. Ogni parte del sistema può essere migliorato utilizzando tecniche più aggiornate e implementazioni competitive.

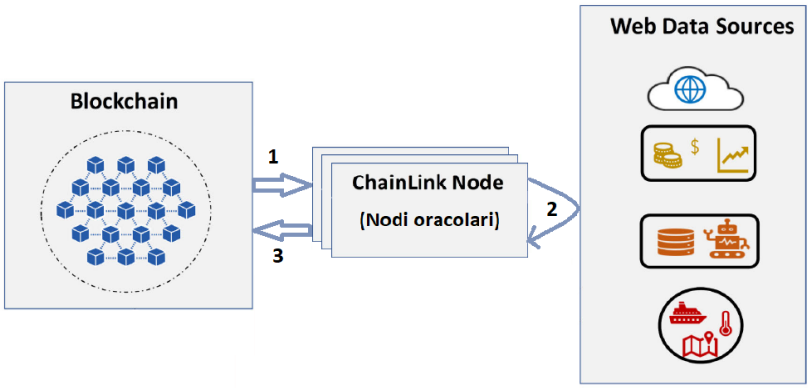

Figura 2.1. Esecuzione del flusso di lavoro

L'esecuzione del flusso di lavoro di Chainlink, rappresentata nella Figura 2.1, è:

1. Uno smart contract all'interno della blockchain richiede dati esterni alla rete,
2. I nodi oracolari, ricevuta la richiesta, interrogano le risorse esterne (API),
3. I nodi oracolari trasmettono i dati ricevuti dalle API alla blockchain, dove verranno aggregati e utilizzati dallo smart contract che ne ha fatto richiesta.

Ciascuna parte del processo è a sua volta costituita da sottoprocessi, ciascuno svolto da una componente specifica dei sistemi on-chain e off-chain.

2.1 Architettura on-chain

L'architettura on-chain di Chainlink si compone di due tipologie di smart contract:

- USER-SC rappresenta un contratto utente. Esso è responsabile dei requesting contracts, ovvero quei contratti che richiedono dati da fonti esterne alla blockchain, in cui viene specificato il tipo di dati richiesti e le condizioni che devono essere soddisfatte affinché i dati siano considerati validi. Nello specifico, USER-SC deve creare una richiesta indicando l'indirizzo dell'oracolo, l'ID del lavoro in modo che l'oracolo sappia quali compiti eseguire e la funzione di callback a cui l'oracolo invia la risposta.
- CHAINLINK-SC rappresenta un contratto on-chain relativo all'interfaccia per la richiesta di contratti. Esso è responsabile di tre tipi di contratto:
  - *reputation contract*,
  - *order-matching contract*,
  - *aggregating contract*.

  Il primo tipo di contratto è progettato per misurare e tracciare la reputazione di un particolare utente o entità su una rete blockchain sulla base del loro comportamento. Nello specifico, viene assegnato a ciascun fornitore di servizi oracolo un valore sulla base di diversi fattori, come il tempo di risposta medio, il rapporto di completamento, il deposito di sicurezza medio e altro. Le prestazioni storiche degli oracoli di Chainlink sono pubblicamente disponibili tramite dati firmati sulla catena, consentendo agli utenti di selezionare gli oracoli in base a tali metriche, garantendo fiducia e credibilità di tale fornitore. Gli operatori dei nodi hanno anche la possibilità di fornire dati aggiuntivi come la loro identità, la posizione geografica e le certificazioni di terze parti.

  Il secondo tipo di contratto è progettato per prendere e registrare una proposta di accordo sul livello di servizio (*service level agreement*, con acronimo SLA) e relativi parametri, raccogliere le offerte dei fornitori di oracolo, selezionare le offerte utilizzando il *reputation contract* e finalizzare lo SLA.

  Il terzo tipo di contratto è progettato per raccogliere le risposte dei fornitori di oracoli e calcolare il risultato collettivo finale della query. Esso, inoltre, è responsabile dell'alimentazione delle metriche dei fornitori di oracoli nel *reputation contract*.

I contratti di ChainLink sono progettati in modo modulare, in modo da permettere agli utenti di configurarli o sostituirli a seconda delle necessità. Ciò che avviene all'interno dell'architettura on-chain si articola in tre fasi:

1. appena risulta necessario richiedere dati esterni alla blockchain, viene selezionato l'oracolo (o rete di oracoli) a cui affidare il compito,
2. trasmissione dei dati all'interno della rete da parte di ciascun oracolo,
3. aggregazione dei risultati.

Durante la prima fase, un acquirente di servizi oracolo specifica i requisiti che costituiscono una proposta SLA, includendo dettagli quali i parametri della query e il numero di oracoli necessari. Inoltre, l'acquirente specifica il *reputation contract* e l'*aggregation contract* da utilizzare per il resto dell'accordo [1]. In tal modo, Chainlink consente agli utenti di definire i termini del lavoro di oracolo richiesto negli on-chain smart contracts. È anche possibile richiedere che i nodi oracolo versino un deposito di sicurezza che viene restituito al nodo solo se esegue il lavoro secondo i

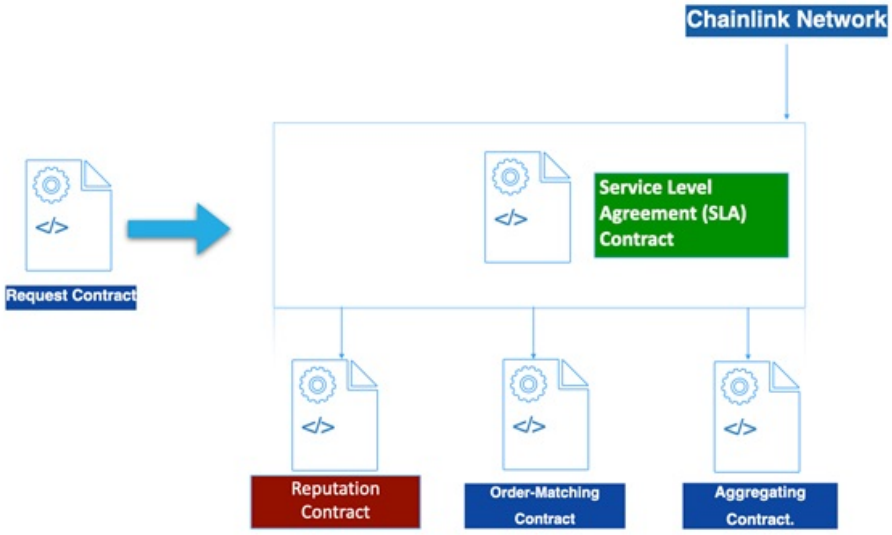

Figura 2.2. Smart contracts

termini preconcordati (ad esempio, i dati vengono consegnati in tempo). La penale depositata è in LINK, il token nativo della piattaforma Chainlink, e incentiva gli oracoli a fornire dati accurati e affidabili.

Utilizzando i registri dei contratti passati e le prestazioni presenti nella catena vengono generati elenchi off-chain, da cui gli acquirenti possono ordinare, filtrare e selezionare manualmente gli oracoli. Quando la selezione manuale è possibile, agli oracoli selezionati viene presentata la proposta SLA e viene raggiunto un accordo tra acquirente e oracoli prima di finalizzare lo SLA sulla catena. Quando invece c'è dinamicità nelle richieste da un contratto ai nodi oracolari, invece di contattare direttamente gli oracoli, l'acquirente sottopone la propria proposta SLA a un *order-matching contract*. Viene generato un registro che i fornitori di oracoli possono monitorare e filtrare in base alle loro capacità e ai loro obiettivi di servizio. Se un oracolo sceglie di fare un'offerta per la proposta, si impegna a rispettare il contratto in quanto allega la penale che andrebbe persa nel caso di un suo comportamento scorretto, come definito nello SLA.

Tra le offerte effettuate, il contratto accetta solo quelle relative ai nodi che soddisfano i requisiti dello SLA per tutta la durata del periodo di offerta. Al termine di tale periodo e se il numero di offerte che soddisfano i requisiti ha raggiunto il numero di oracoli richiesto dall'acquirente, i nodi oracolari vengono selezionati tra tutti quelli validi. Agli oracoli non selezionati viene restituita la penale e viene registrato nella catena lo SLA definitivo. Si attiva un registro che notifica gli oracoli selezionati e quest'ultimi eseguono l'incarico.

Gli oracoli raccolgono e selezionano i dati richiesti dal mondo esterno alla blockchain e li trasmettono nuovamente all'interno dell'architettura on-chain. Questo passaggio di dati costituisce la seconda fase del flusso di lavoro all'interno della catena.

Durante la terza fase, invece, i dati ricevuti dagli oracoli vengono confrontati dall'*aggregating contract*, il quale calcola una risposta ponderata che viene restituita al contratto specifico in USER-SC. Inoltre, la validità di ogni risposta dell'oracolo viene comunicata al *reputation contract*. Non esiste un unico *aggregating contract*, ma un insieme standard di tali contratti, in quando la scelta del contratto e l'identificazione di outliers o valori errati dipendono dall'applicazione e dal tipo di dati con cui si ha a che fare (ad esempio dati numerici o booleani). Dunque, l'acquirente specifica l'indirizzo di contratto configurabile tra quelli esistenti. Inoltre, è possibile specificare anche contratti personalizzati, purché conformi all'interfaccia di calcolo standard.

2.2 Architettura off-chain

L'architettura off-chain di Chainlink è composta da nodi oracolari connessi alla blockchain. Ciascun oracolo opera senza alcuna dipendenza da altri oracoli, ovvero raccoglie le risposte dalle richieste off-chain in modo indipendente. Inoltre, può liberamente far parte contemporaneamente di diverse reti di oracoli.

Il tipo di lavoro più comune per un nodo consiste nell'effettuare una richiesta GET a un'API (ovvero un metodo di richiesta HTTP per prelevare dati da una fonte), recuperare alcuni dati da essa, analizzare la risposta, convertire il risultato in dati compatibili con la blockchain e quindi inviarlo in una transazione al contratto oracolo. Le singole risposte pervenute dai diversi nodi oracolari vengono combinate on-chain tramite uno dei diversi meccanismi di consenso disponibili, per produrre una risposta globale che viene restituita a un *requesting contract* USER-SC.

Ciascun nodo oracolo è caratterizzato dalla massima flessibilità per quanto riguarda i tipi di dati che possono essere recuperati e il modo in cui tali dati possono essere forniti. Infatti, ogni nodo dispone di una serie di *core adapters* precostruiti, che gli consentono di connettersi a qualsiasi API aperta e di fornire i dati sulla catena. Questi adattatori forniscono ai nodi Chainlink alcune funzionalità, ma è possibile aggiungere estensioni software, note come *external adapters*, che offrono ulteriori servizi specializzati off-chain e permettono di accedere a qualunque risorsa esterna. Tali servizi riguardano principalmente la varietà di dati a cui accedere e il tipo di calcoli che possono essere eseguiti. Gli *external adapters* possono, ad esempio, eseguire calcoli off-chain sui dati (producendo una media delle risposte dei nodi) o accedere ad API autenticate che richiedono credenziali.

La differenza sostanziale tra le due tipologie di nodi oracolari è la seguente:

- i nodi *Core* si interfacciano con gli smart contracts della blockchain e si occupano di pianificare e bilanciare il lavoro tra i vari servizi esterni,
- i nodi *External Adapters* comunicano con le API esterne con una semplice specifica JSON e sono la rappresentazione off-chain di servizi REST-API.

Il lavoro eseguito dai nodi Chainlink viene suddiviso in incarichi. Ciascun incarico è costituito da un insieme di sotto-attività più piccole, che vengono elaborate come una pipeline. Infatti, ogni sotto-attività è responsabile di eseguire un'operazione specifica e trasmettere il risultato alla sotto-attività successiva, e così via fino a raggiungere un risultato finale. Alcune sotto-attività sono integrate nel software del nodo di ChainLink, tra cui le richieste HTTP, JSON *pairing* e la conversione in vari formati di blockchain.

Oltre alle sotto-attività integrate, è possibile definire sotto-attività personalizzate creando gli *adapters*. Modellando tali adattatori in modo orientato ai servizi, è possibile implementare con facilità i programmi in qualsiasi linguaggio di programmazione aggiungendo una piccola API intermedia nella parte iniziale del programma. Allo stesso modo, è possibile semplificare l'interazione con complicate API multi-step utilizzando singole sotto-attività parametrizzate. Le informazioni sugli adattatori esterni sono suddivise in tre categorie principali:

- creatori di contratti, responsabili di specificare la richiesta di dati esterni;
- sviluppatori, responsabili di implementare un adattatore esterno per un'API;
- operatori di nodi, responsabili di aggiungere un adattatore esterno al proprio nodo, in modo da poter fornire servizi specializzati agli smart contracts.

Per via degli *external adapters*, la rete Chainlink può espandersi continuamente per supportare nuove funzionalità, come la comunicazione bidirezionale, i pagamenti bancari off-chain, l'interoperabilità con altre blockchain e molto altro, senza mettere a rischio le funzioni principali della rete.

La maggior parte degli *adapters* sono open source, per cui vari membri della comunità possono verificare ed eseguire i diversi servizi. Poiché ci sono molti tipi di adattatori creati da numerosi sviluppatori diversi, è fondamentale assicurarsi che gli adattatori siano compatibili tra loro. L'idea iniziale dei creatori di Chainlink è quella di utilizzare un sistema di schemi basato sullo schema JSON, un linguaggio dichiarativo che consente di annotare e convalidare documenti JSON. In tal modo è possibile specificare quali sono gli input di cui ogni adattatore ha bisogno e come devono essere formattati. Allo stesso modo, gli adattatori specificano uno schema di output per descrivere il formato dell'output di ogni sotto-attività.

Dopo aver specificato tutti i componenti dell'architettura di Chainlink, il flusso di lavoro più dettagliato, descritto dalla Figura 6.1, è il seguente:

1. USER-SC effettua una richiesta on-chain,
2. CHAINLINK-SC registra un evento per gli oracoli,
3. il nodo *Core* di ChainLink raccoglie l'evento e instrada l'assegnazione a un *adapter*,
4. l'*adapter* di ChainLink esegue una richiesta a un'API esterna
5. l'*adapter* elabora la risposta e la passa di nuovo al *Core*,
6. il *Core* riporta i dati a CHAINLINK-System,
7. CHAINLINK-SC aggrega le risposte e trasmette la singola risposta a USER-SC.

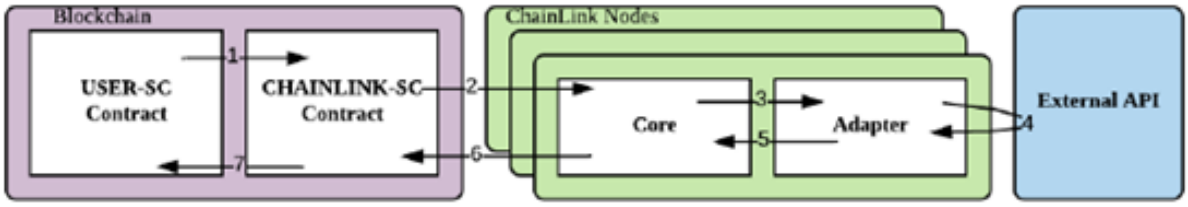

Figura 2.3. Flusso di lavoro ad alto livello

Capitolo 3

Sicurezza dell'oracolo

La sicurezza degli oracoli è un aspetto fondamentale per garantire l'affidabilità e l'integrità delle informazioni fornite agli smart contract basati sulla tecnologia blockchain. Prima di spiegare l'architettura di sicurezza di ChainLink, ovvero i meccanismi che mette in atto, verrà spiegato più nel dettaglio il motivo per cui la sicurezza è importante e cosa significa.

Perché gli oracoli devono essere sicuri?
Per ciascuno dei tre esempi presentati nell'introduzione si possono considerare le seguenti carenze in ambito di sicurezza:

- se un titolo di smart contract riceve un feed di dati falso, potrebbe pagare la parte errata,
- se i feed di dati dell'assicurazione di smart contract possono essere manomessi dalla parte assicurata, potrebbe esserci una frode assicurativa,
- se i dati GPS relativi a una spedizione forniti a un contratto di finanziamento commerciale possono essere modificati dopo che lasciano il provider, il pagamento può essere erogato per le merci che non sono arrivate.

Più in generale, una blockchain ben funzionante offre proprietà di sicurezza molto forti. Gli utenti si affidano alla blockchain come una funzionalità che convalida correttamente le transazioni e impedisce ai dati di essere alterato. Lo trattano a tutti gli effetti come una terza parte fidata. Un servizio di oracoli di supporto alla blockchain deve offrire un livello di sicurezza adeguato con quello della blockchain stessa. Anche un oracolo deve quindi servire gli utenti come un'efficace terza parte fidata, fornendo risposte corrette e tempestive con altissima probabilità. La sicurezza di qualsiasi sistema è forte solo quanto lo è il suo anello più debole, quindi è necessario un oracolo altamente affidabile per preservare l'affidabilità di una blockchain ben progettata.

3.1 Oracolo ideale

Per ragionare sulla sicurezza dell'oracolo, è necessario prima definirla. Un modo istruttivo e basato sui principi di ragionare sulla sicurezza dell'oracolo deriva dal seguente esperimento mentale. Si può immaginare che una terza parte fidata (TTP), che sia un'entità o una funzionalità ideale che esegue sempre le istruzioni fedelmente alla lettera, abbia il compito di gestire un oracolo. Tale oracolo verrà indicato con ORACLE (utilizzando tutte le maiuscole in generale per indicare un'entità completamente fidata dagli utenti) e si suppone che TTP ottenga i dati da una fonte di dati perfettamente affidabile *Src*. Dato questo servizio magico ORACLE, quali istruzioni è

possibile chiedergli di eseguire? Per ottenere la proprietà di **integrità**, nota anche come proprietà di **autenticità**, verrà chiesto semplicemente all'ORACLE di eseguire i seguenti passaggi:

1. *Accettare la richiesta*: importare da uno smart contract USER-SC una richiesta $Req = (Src, \tau, q)$ che specifica un'origine dati di destinazione Src, un'ora o un intervallo di volte τ e una richiesta q;
2. *Ottenere i dati*: inviare la richiesta q a Src all'istante τ;
3. *Restituire i dati*: alla ricezione della risposta a, restituire a allo smart contract.

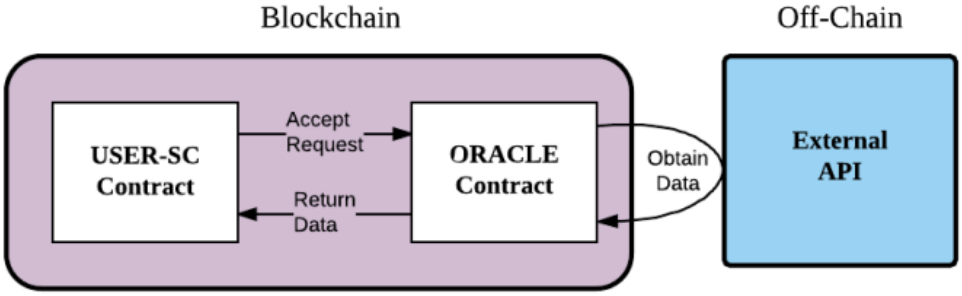

Figura 3.1. Comportamento di un oracolo ideale

Queste semplici istruzioni, eseguite correttamente, definiscono una nozione di sicurezza forte, significativa, ma semplice. Intuitivamente, impongono che ORACLE agisca come un ponte affidabile tra Src e USER-SC. Ad esempio, se Src è il sito web "$https://www.FountOfKnowledge.com$", τ corrisponde alle 16:00 e q = "prezzo per ticker INTC", l'integrità dell'ORACLE garantisce che fornirà a USER-SC esattamente il prezzo di INTC come richiesto alle 16:00 al sito web "$https://www.FountOfKnowledge.com$".

La **riservatezza** è un'altra proprietà desiderabile per gli oracoli. Poiché USER-SC invia Req all'ORACLE in chiaro sulla blockchain, Req è pubblico. Ci sono molte situazioni in cui Req è sensibile e la sua pubblicazione potrebbe essere dannosa. Se USER-SC è un contratto di assicurazione per voli aerei, ad esempio, e invia all'ORACLE una richiesta Req riguardante un volo di un particolare utente (q = "Ether Air Flight 338"), il risultato sarebbe che i piani di volo di un utente vengono rivelati al mondo intero. Se, invece, USER-SC è un contratto per il trading finanziario, Req potrebbe far trapelare informazioni sulle operazioni e sul portafoglio di un utente. Ci sono molti altri esempi, ovviamente. Per proteggere la riservatezza di Req, possiamo richiedere che i dati in Req siano crittografati sotto una (chiave pubblica) appartenente all'ORACLE. Continuando a sfruttare la natura TTP dell'ORACLE, si può quindi semplicemente assegnare all'ORACLE il vincolo del flusso di informazioni. Inoltre, una volta decifrato Req, non bisogna rivelare o utilizzare mai i dati in Req se non per interrogare Src.

Ci sono altre importanti proprietà dell'ORACLE, come la **disponibilità**, l'ultima delle classica triade CIA (*Confidentiality-Integrity-Availability*, ovvero Riservatezza-Integrità-Disponibilità). Un servizio davvero ideale dell'ORACLE, ovviamente, non andrebbe mai in crisi. La disponibilità comprende anche proprietà più sottili come la resistenza alla censura: un oracolo onesto non individuerà particolari contratti intelligenti e negherà le loro richieste. Il concetto di una terza parte fidata è simile alla nozione di una funzionalità ideale utilizzata per dimostrare la sicurezza dei

protocolli crittografici in alcuni modelli. Si può anche modellare una blockchain in termini simili, concettualizzandola in termini di TTP che mantiene una bacheca ideale. Le sue istruzioni sono di accettare transazioni, convalidarli, serializzarli e mantenerli permanentemente sulla bacheca, una struttura di dati di sola aggiunta.

Perché l'oracolo ideale è difficile da raggiungere.
Ovviamente non esiste una fonte di dati *Src* perfettamente affidabile. I dati possono essere danneggiati in modo benigno o dannoso a causa di siti web difettosi, fornitori di servizi imbroglioni o errori onesti. Se *Src* non è affidabile, anche se ORACLE funziona esattamente come un TTP come indicato sopra, non soddisfa ancora completamente la nozione di sicurezza necessaria. Data una sorgente errata *Src*, la proprietà di integrità definita sopra non significa più che la risposta a di un oracolo è corretta. Se il vero prezzo di Intel è 40 e "$https://www.FountOfKnowledge.com$" lo riporta erroneamente come 50, ad esempio, l'oracolo invierà il valore errato a = 50 a USER-SC. Questo problema è inevitabile quando si utilizza un'unica sorgente *Src*. L'oracolo semplicemente non ha modo di sapere se le risposte che *Src* fornisce alle sue domande sono corrette. Un problema più grande, ovviamente, è il fatto che il nostro TTP per ORACLE è solo un'astrazione. Nessun fornitore di servizi è incondizionatamente affidabile. Anche il più intenzionato potrebbe essere difettoso o violato. Quindi non c'è modo per un utente o un contratto intelligente di avere l'assoluta certezza che un servizio ORACLE eseguirà fedelmente le sue istruzioni.

ChainLink ragiona sui suoi protocolli di sicurezza in termini di questa funzionalità ideale ORACLE. L'obiettivo in ChainLink è realizzare un sistema del mondo reale con proprietà il più vicino possibile a quelli di ORACLE sotto ipotesi realistiche di fiducia. Verrà spiegato successivamente come questo sia possibile. Per semplicità in quanto segue, verrà indicato con CHAINLINK-SC l'insieme completo di contratti ChainLink, ovvero la sua piena funzionalità on-chain (non solo la sua interfaccia per la richiesta di contratti). In questo modo vengono estratti i molteplici contratti individuali effettivamente utilizzati nell'architettura del sistema.

Capitolo 4

Approccio alla decentralizzazione di ChainLink

Si possono proporre tre soluzioni complementari che garantiscono la sicurezza e l'affidabilità della rete Chainlink contro i nodi difettosi:

1. distribuzione delle fonti di dati;

2. distribuzione di oracoli;

3. utilizzo di hardware affidabile.

Tali soluzioni possono essere applicati singolarmente o combinati tra loro, aumentando in tal modo l'attendibilità dei dati. In particolare, i primi due approcci sono degli esempi di decentralizzazione della rete. Pertanto, verranno discussi in questa sezione. Nella figura 4.1 viene mostrato un esempio in cui questi due approcci vengono usati in modo combinato.

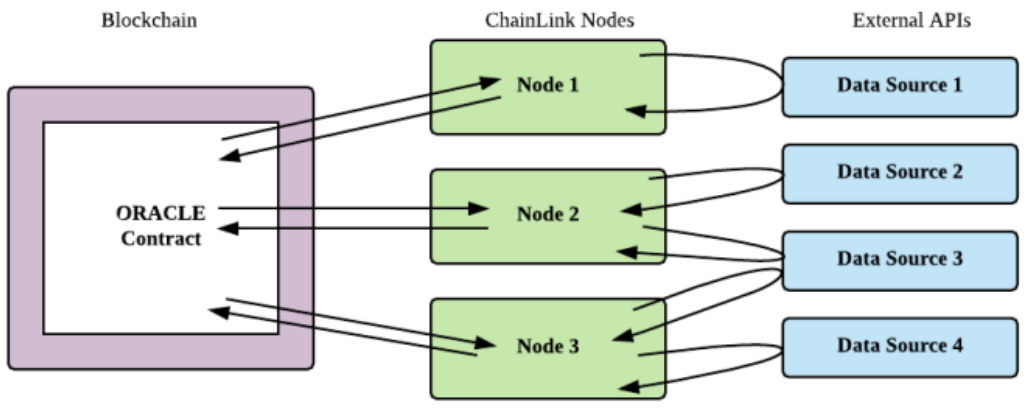

Figura 4.1. Decentralizzazione del Chainlink

4.1 Fonti distributive

Un modo semplice per gestire una singola fonte di dati difettosa Src consiste nell'ottenere dati da più fonti, ovvero distribuire l'origine dati. Un ORACLE affidabile può interrogare una raccolta delle sorgenti $Src_1, Src_2, ..., Src_k$, ottenere le risposte $a_1, a_2, ..., a_k$, e infine aggregarle in un'unica risposta $A = agg(a_1, a_2, ..., a_k)$.

L'aggregazione dei risultati potrebbe essere effettuata in una miriade di modi diversi. Uno dei modi più comuni, ad esempio, è il voto a maggioranza. Se la maggior parte delle fonti restituisce lo stesso valore a, la funzione agg restituisce tale valore; altrimenti restituisce un errore. In questo caso, a condizione che la maggioranza ($> k/2$) delle fonti funzioni correttamente, ORACLE restituirà sempre un valore corretto A.

Molte funzioni alternative agg possono garantire robustezza contro dati errati o gestire le fluttuazioni dei valori dei dati nel tempo (ad esempio, i prezzi delle azioni). In tal caso, agg potrebbe, ad esempio, scartare valori anomali (come i valori più grandi e più piccoli a_i) e produrre la media di quelli rimanenti.

Tuttavia, si può presentare un ulteriore problema. Gli errori possono essere correlati tra le fonti di dati e ciò comporta l'indebolimento delle garanzie fornite dall'aggregazione. Se il sito $Src_1 = EchoEcho.com$ ottiene i suoi dati da $Src_2 = TheHorsesMouth.com$, un errore in Src_2 implicherà sempre un errore in Src_1. Possono verificarsi anche correlazioni più sottili tra le fonti di dati. Chainlink propone inoltre di proseguire la ricerca sulla mappatura e la segnalazione dell'indipendenza delle fonti di dati in un modo facilmente digeribile in modo che gli oracoli e gli utenti possano evitare correlazioni indesiderate.

4.2 Oracoli distributivi

Proprio come le fonti possono essere distribuite, anche il servizio di oracolo ideale può essere approssimato come un sistema distribuito. Questo equivale a dire che invece di un singolo nodo oracolo monolitico O, si possono invece avere una raccolta di n diversi nodi oracolo $O_1, O_2, ..., O_n$. Ogni oracolo O_i contatta il proprio insieme distinto di fonti di dati che possono o meno sovrapporsi a quelle di altri oracoli. O_i aggrega le risposte dalle sue origini dati e restituisce la propria risposta distinta A_i a una query Req.

Alcuni di questi oracoli potrebbero essere errati. Quindi, chiaramente, l'insieme di tutte le risposte degli oracoli $A_1, A_2, ..., A_n$ dovrà essere aggregato in modo affidabile in un singolo valore autorevole A. Ma data la possibilità di oracoli difettosi, dove e come avverrà questa aggregazione in ChainLink?

Soluzione iniziale: aggregazione in-contract. La soluzione iniziale proposta in ChainLink è stata chiamata *aggregazione in-contract*. CHAINLINK-SC che, ancora una volta, denota la parte sulla catena di ChainLink, aggregherà a sua volta le risposte dell'oracolo. (In alternativa, CHAINLINK-SC può chiamare un altro contratto di aggregazione, ma per semplicità concettuale viene assunto che le due componenti formino un unico contratto.) In altre parole, CHAINLINK-SC calcolerà $A = Agg(A_1, A_2, ..., A_n)$ per qualche funzione Agg (simile a agg, come descritto sopra) e invierà il risultato A a USER-SC. Questo approccio è pratico per n piccoli e presenta diversi vantaggi distinti:

- *Semplicità computazionale*: Nonostante il fatto che l'oracolo sia distribuito, una singola entità, CHAINLINK-SC, esegue l'aggregazione eseguendo Agg.

- *Affidabilità*: Poiché il codice di CHAINLINK-SC può essere ispezionato pubblicamente, è possibile verificarne il corretto comportamento. (CHAINLINK-SC sarà un pezzo di codice relativamente piccolo e semplice.) Inoltre, l'esecuzione di CHAINLINK-SC è completamente visibile on-chain. Così gli utenti, cioè i creatori di USER-SC, possono raggiungere un alto grado di fiducia in CHAINLINK-SC.

- *Flessibilità*: CHAINLINK-SC può implementare le funzioni aggregate *Agg* più desiderate, come la funzione di maggioranza, la media, ecc.

Per quanto semplice, questo approccio presenta una nuova e interessante sfida tecnica, vale a dire il problema del freeloading. Un oracolo imbroglione O_z può osservare la risposta A_i di un altro oracolo O_i e copiarla. In questo modo, l'oracolo O_z evita la spesa di interrogare le fonti di dati, che possono addebitare tariffe per query. Dunque, il freeloading indebolisce la sicurezza minando la diversità delle query sull'origine dei dati e inoltre disincentiva gli oracoli dal rispondere rapidamente: rispondere lentamente combinata con il freeloading è una strategia più economica.

Una soluzione a questo problema corrisponde all'uso di uno schema *commit/reveal*. In un primo round, gli oracoli inviano a CHAINLINK-SC impegni crittografici alle loro risposte. Dopo che CHAINLINK-SC ha ricevuto un quorum di risposte, avvia un secondo round in cui gli oracoli rivelano le loro risposte. In [1] è stato proposto un algoritmo, che mostra un semplice protocollo sequenziale che garantisce la disponibilità data $3f + 1$ nodi. Utilizza uno schema di commit/rivelazione per impedire il freeload. Le risposte degli oracoli vengono disattivate e quindi esposte a un potenziale freeloader solo dopo che tutti gli impegni sono stati presi, escludendo così il freeloader dalla copia di altre risposte degli oracoli. I protocolli on-chain possono sfruttare i tempi di blocco per supportare progetti di protocolli sincroni. In ChainLink, tuttavia, i nodi oracolari ottengono dati da fonti che possono avere tempi di risposta molto variabili e i tempi di disattivazione dei nodi possono variare a causa di, ad esempio, utilizzo di diversi prezzi del gas in Ethereum. Per garantire la reattività del protocollo più veloce possibile, quindi, l'algoritmo proposto è progettato come protocollo asincrono. Il protocollo presuppone canali autenticati tra tutti i giocatori. Inoltre, viene garantito con tale algoritmo che la risposta finale aggregata sarà corretta, se si suppone che i nodi difettosi siano al massimo f. Dati $3f + 1$ nodi in totale, almeno $2f + 1$ invieranno impegni. Di quegli impegni, al massimo f provengono da nodi difettosi, quindi almeno $f + 1$ provengono da nodi onesti. Tutti questi impegni saranno infine disimpegnati. Dei $f + 1$ disimpegni sul singolo valore A, almeno uno deve provenire da un nodo onesto.

L'*aggregazione in-contract* sarà l'approccio principale supportato da ChainLink a breve termine. L'implementazione iniziale proposta comporterà una variante più sofisticata e simultanea dell'algoritmo. La proposta a più lungo termine si riflette nel protocollo OCA (Off-Chain Aggregation) piuttosto complicato. OCA è un protocollo di aggregazione off-chain che riduce al minimo i costi di transazione on-chain. Tale protocollo include anche il pagamento ai nodi oracolari e garantisce contro i pagamenti ai freeloader.

Strategia a medio termine: aggregazione off-chain. L'*aggregazione in-contract* ha uno svantaggio chiave: il costo. Sostiene il costo della trasmissione e dell'elaborazione dei messaggi Oracle sulla catena $O(n)$ (commit e rivelazioni per $A_1, A_2, ..., A_n$). In blockchain permissioned, questo sovraccarico può essere accettabile. Nelle blockchain permissionless con commissioni di transazione on-chain come Ethereum, se n è grande, i costi possono essere proibitivi. Un un approccio più conveniente consiste nell'aggregare le risposte degli oracoli off-chain e trasmettere un singolo messaggio A a CHAINLINK-SC. Questo approccio è chiamato *aggregazione off-chain* e viene proposto nel medio-lungo termine. Il problema di raggiungere un valore di consenso A di fronte a nodi potenzialmente difettosi è molto simile al problema del consenso che è alla base

delle stesse blockchain. Dato un insieme predeterminato di oracoli, si potrebbe considerare l'utilizzo di un algoritmo di consenso classico Byzantine Fault Tolerant (BFT) per calcolare A. I protocolli BFT classici, tuttavia, mirano a garantire che alla fine di un'invocazione di protocollo, tutti i nodi onesti memorizzino lo stesso valore, ovvero che tutti i nodi memorizzano lo stesso nuovo bloccoin una blockchain. In questa impostazione dell'oracolo, l'obiettivo è leggermente diverso. Si vuole assicurare che CHAINLINK-SC (e quindi USER-SC) ottiene la risposta aggregata $A = Agg(A_1, A_2, ..., A_n)$ senza partecipare al protocollo di consenso e senza bisogno di ricevere risposte da più oracoli. Il problema del freeloading, inoltre, deve ancora essere affrontato.

ChainLink propone l'uso di un semplice protocollo che coinvolge le firme di soglia. Tali firme possono essere realizzate utilizzando uno qualsiasi di un certo numero di schemi di firma, ma sono particolarmente semplici da implementare utilizzando le firme di Schnorr. In questo approccio, gli oracoli hanno una chiave pubblica collettiva pk e una corrispondente chiave privata sk condivisa tra $O_1, O_2, ..., O_n$ in modo (t, n)-soglia. Tale condivisione significa che ogni nodo O_i ha una distinta coppia di chiavi privata/pubblica (sk_i, pk_i). O_i può generare una firma parziale $\sigma_i = Sig_{sk_i}[A_i]$ verificabile rispetto a pk_i. La caratteristica chiave di questa configurazione è che le firme parziali sullo stesso valore A possono essere aggregate attraverso qualsiasi insieme di oracoli per produrre un'unica firma collettiva valida $\Sigma = Sig_{sk}[A]$ su una risposta A. Nessun insieme di $t-1$ oracoli, tuttavia, può produrre una firma valida su qualsiasi valore. La firma singola Σ incorpora quindi implicitamente le firme parziali di almeno t oracoli. Le firme di soglia possono essere realizzate in modo ingenuo lasciando che Σ consista esplicitamente di un insieme di t firme indipendenti valide dai singoli nodi. Le firme di soglia hanno proprietà di sicurezza simili a questo approccio ingenuo, ma forniscono un significativo miglioramento delle prestazioni on-chain: riducono le dimensioni e il costo della verifica di Σ di un fattore di t. Con questa configurazione, sembrerebbe che gli oracoli possano solo generare e trasmettere firme parziali finchè t di tali firme parziali consentono il calcolo di Σ.

Di nuovo, però, si pone il problema del freeloading. Bisogna quindi garantire che gli oracoli ottengano realmente i dati dalle loro fonti designate, piuttosto che barare e copiare A_i da un altro oracolo. La soluzione proposta prevede un meccanismo finanziario: un'entità FORNITORE (realizzabile come smart contract) premia solo gli oracoli che hanno fornito dati originali per le loro firme parziali. In un ambiente distribuito, determinare quali oracoli si qualificano per il pagamento risulta complicato. Gli oracoli possono intercomunicare off-chain e, non essendoci più un'unica entità autorevole (CHAINLINK-SC) che riceve risposte, pertanto non si è più in grado di identificare i beneficiari idonei direttamente tra gli oracoli partecipanti. Di conseguenza, il FORNITORE deve ottenere prove di comportamento scorretto dagli stessi oracoli, alcuni dei quali potrebbero essere inaffidabili. Viene proposto l'uso di meccanismi simili al consenso al fine di garantire che il FORNITORE non paghi gli oracoli che applicano il freeloading. Nello specifico, l'algoritmo proposto fa uso di un protocollo distribuito basato su firme di soglia che fornisce resistenza al freeloading da parte di $f < n/3$ oracoli.

Capitolo 5

Servizi di Sicurezza ChainLink

Grazie ai protocolli descritti nella sezione precedente, ChainLink si propone di garantire disponibilità e correttezza fino al caso in cui ci siano **f** oracoli difettosi. Inoltre, l'hardware fidato viene considerato attivamente come approccio sicuro alla protezione da oracoli corrotti che forniscono risposte errate. L'hardware affidabile, tuttavia, potrebbe non fornire una protezione definitiva per tre motivi. Innanzitutto, non verrà distribuito nelle versioni iniziali della rete ChainLink. In secondo luogo, alcuni utenti potrebbero non fidarsi dell'hardware affidabile. Infine, l'hardware affidabile non può proteggere dai tempi di inattività del nodo, ma solo dal comportamento scorretto del nodo. Gli utenti vorranno quindi assicurarsi di poter scegliere gli oracoli più affidabili e ridurre al minimo la probabilità che USER-SC faccia affidamento su oltre f oracoli difettosi. A tal fine, è stato proposto l'utilizzo di quattro servizi di sicurezza chiave: un sistema di convalida, un sistema di reputazione, un servizio di certificazione e un servizio di aggiornamento del contratto. I primi tre servizi forniscono solo valutazioni o indicazioni agli utenti, mentre il servizio di aggiornamento dei contratti è del tutto facoltativo per gli utenti. Tutti questi servizi possono inizialmente essere gestiti da un'azienda o da un gruppo interessato a lanciare la rete ChainLink, ma sono progettati per operare rigorosamente in conformità con gli obiettivi di ChainLink. Inoltre, essi non possono bloccare la partecipazione dei nodi oracolo o alterarne le risposte.

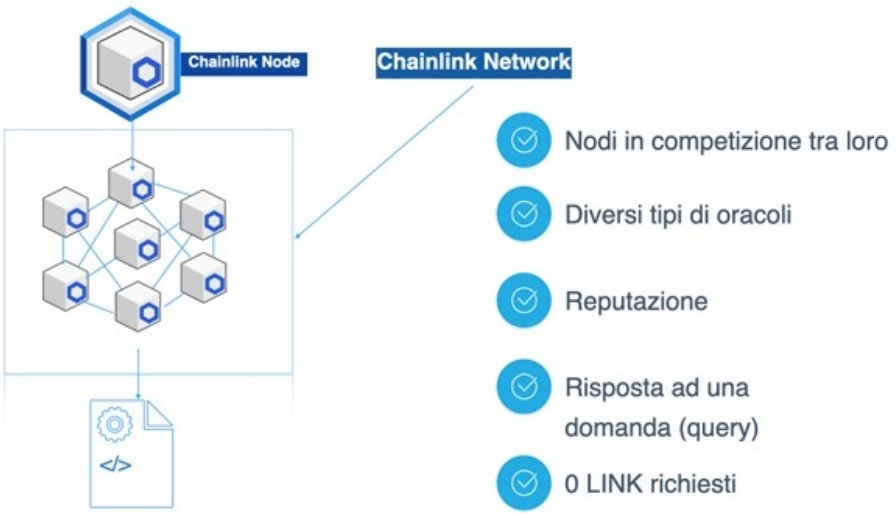

Figura 5.1. Come Chainlink garantisce la sicurezza

5.1 Sistema di Validazione

Il sistema di convalida ChainLink monitora il comportamento degli oracoli sulla catena, fornendo una metrica oggettiva delle prestazioni che può guidare la selezione degli oracoli da parte dell'utente. Cercherà di monitorare gli oracoli per:

- *Disponibilità*: Il sistema di convalida dovrebbe registrare gli errori di un oracolo per rispondere in modo tempestivo alle domande. Compila statistiche di uptime in corso.

- *Correttezza*: Il sistema di convalida dovrebbe registrare risposte errate apparenti da parte di un oracolo misurate in base alle deviazioni dalle risposte fornite dai colleghi.

Nel sistema iniziale di aggregazione on-chain in ChainLink, tale monitoraggio è semplice, poiché tutta l'attività dell'oracolo è visibile a CHAINLINK-SC. Tuttavia, nel sistema di aggregazione off-chain previsto per ChainLink, sono gli oracoli stessi che eseguono l'aggregazione. Di conseguenza, CHAINLINK-SC non ha visibilità diretta sulle risposte degli oracoli e non può monitorare la disponibilità e la correttezza. Fortunatamente, gli oracoli firmano digitalmente le loro risposte e quindi, come effetto collaterale, generano prove non ripudiabili delle loro risposte. L'approccio proposto sarà quindi quello di realizzare il servizio di convalida come un contratto intelligente che premierebbe gli oracoli per la presentazione di prove di risposte divergenti. In altre parole, gli oracoli sarebbero incentivati a segnalare comportamenti apparentemente errati. La disponibilità è in qualche modo più complicata da monitorare, poiché gli oracoli ovviamente non firmano mancate risposte. Invece, un miglioramento del protocollo proposto richiederebbe agli oracoli di firmare digitalmente le attestazioni per l'insieme di risposte che hanno ricevuto da altri oracoli. Il contratto di convalida accetterebbe quindi (e ricompenserebbe nuovamente) l'invio di insiemi di attestazioni che dimostrano una costante non reattività da parte di un oracolo con prestazioni insufficienti nei confronti dei suoi pari. In entrambi i casi on-chain e off-chain, le statistiche di disponibilità e correttezza per gli oracoli saranno visibili sulla catena. Gli utenti/sviluppatori potranno così visualizzarli in tempo reale attraverso un front end appropriato, come una Dapp in Ethereum o un'applicazione equivalente per una blockchain permissioned.

5.2 Sistema di Reputazione

Il sistema di reputazione proposto per ChainLink registrerebbe e pubblicherebbe le valutazioni degli utenti di provider e nodi Oracle, offrendo agli utenti un mezzo per valutare le prestazioni di Oracle olistico. È probabile che i rapporti del sistema di convalida siano un fattore importante nella determinazione delle reputazioni oracolari e che ponngano queste reputazioni su una solida base di fiducia. Fattori al di là della cronologia on-chain, tuttavia, possono fornire informazioni essenziali sui profili di sicurezza del nodo oracolo. Questi possono includere la familiarità degli utenti con i marchi degli oracoli, entità operative e architetture. Viene previsto che il sistema di reputazione ChainLink includa un componente on-chain di base in cui le valutazioni degli utenti sarebbero disponibili per altri smart contracts a cui fare riferimento. Inoltre, le metriche sulla reputazione dovrebbero essere facilmente accessibili off-chain, dove è possibile elaborare in modo efficiente e più flessibile grandi quantità di dati.
Per un dato operatore Oracle, il sistema di reputazione viene inizialmente proposto per supportare le seguenti metriche:

- *Numero totale di richieste assegnate*: Il numero totale di richieste passate accettate da un oracolo, sia soddisfatte che non soddisfatte.

- *Numero totale di richieste completate*: Il numero totale di richieste passate soddisfatte da un oracolo. Questo può essere calcolato in media sul numero di richieste assegnate per calcolare il tasso di completamento.
- *Numero totale di richieste accettate*: Il numero totale di richieste che sono state ritenute accettabili calcolando i contratti rispetto alle risposte dei pari. Questo può essere calcolato in media rispetto al totale delle richieste assegnate o completate da ottenere comprensione dei tassi di accuratezza.
- *Tempo medio di risposta*: Sebbene possa essere necessario dare alle risposte dell'oracolo il tempo per la conferma, la tempestività delle loro risposte sarà utile per determinare la tempestività futura. Il tempo medio di risposta viene calcolato in base al completamento richieste.
- *Importo delle penalità*: Se i pagamenti di penalità fossero bloccati per garantire le prestazioni di un operatore di nodo, il risultato sarebbe una metrica finanziaria dell'impegno di un fornitore di oracoli a non impegnarsi in un attacco "exit scam", in cui il fornitore prende i soldi degli utenti e non fornisce servizi. Questa metrica lo farebbe coinvolgere sia una dimensione temporale che finanziaria.

I servizi di alta reputazione sono fortemente incentivati in qualsiasi mercato a comportarsi correttamente e garantire disponibilità e prestazioni elevate. Il feedback negativo degli utenti rappresenterà un rischio significativo per il valore del marchio, così come le sanzioni associate al comportamento scorretto. Di conseguenza, si prevede un circolo virtuoso in cui oracoli ben funzionanti sviluppano una buona reputazione e una buona reputazione genera incentivi per continuare a ottenere alte prestazioni.

5.3 Servizio di certificazione

Mentre i sistemi di convalida e reputazione hanno lo scopo di affrontare un'ampia gamma di comportamenti errati da parte degli oracoli e sono proposti come un modo per garantire l'integrità del sistema nella stragrande maggioranza dei casi, ChainLink può includere anche un meccanismo aggiuntivo chiamato servizio di certificazione. Il suo obiettivo è prevenire e/o rimediare a eventi rari ma catastrofici, in particolare imbrogli in blocco sotto forma di attacchi Sybil e di mirroring.

In particolare, il servizio di certificazione ChainLink cerca di fornire una garanzia generale di integrità e disponibilità, rilevando e aiutando a prevenire mirroring e colludendo i quorum degli oracoli nel breve e medio termine. Il servizio di certificazione rilascia approvazioni di fornitori di oracoli di alta qualità. Come notato sopra, il servizio valuta solo i fornitori a vantaggio degli utenti; non ha lo scopo di dettare la partecipazione o la non partecipazione del nodo oracolo al sistema.

5.4 Servizio di aggiornamento dei contratti

La codifica di smart contracts a prova di bomba è un'attività molto complessa e richiede un'attenta pianificazione e attenzione ai dettagli. Tuttavia, anche se un contratto intelligente è stato programmato in modo corretto, cambiamenti ambientali o errori di codifica possono ancora causare vulnerabilità. Per mitigare questi rischi, si propone un servizio di aggiornamento dei contratti opzionale e sotto il controllo degli utenti. Il Contract-Upgrade Service è stato sviluppato per fornire un ulteriore livello di sicurezza e affidabilità agli smart contract basati sulla tecnologia ChainLink.

In caso di scoperta di vulnerabilità, il Contract-Upgrade Service rende disponibile un nuovo set di contratti oracolo di supporto, che gli smart contract richiedenti appena creati potrebbero migrare. Ciò garantiresce che gli smart contract continuino a funzionare in modo affidabile e sicuro, anche in presenza di cambiamenti ambientali o di altri fattori che potrebbero influenzare il loro funzionamento. Tuttavia, l'uso del Contract-Upgrade Service è completamente facoltativo e sotto il controllo degli utenti. Gli utenti possono scegliere di utilizzare il servizio o di continuare a utilizzare i loro contratti originali, se lo desiderano. Inoltre, il servizio viene fornito con una serie di controlli e bilanciamenti per garantire la massima sicurezza possibile, tra cui l'audit indipendente dei nuovi contratti oracolo e l'utilizzo di meccanismi di incentivazione per premiare gli sviluppatori per l'aderenza alle linee guida stabiliti.

5.5 Utilizzo del token LINK

La rete ChainLink utilizza il token LINK per pagare gli operatori del nodo ChainLink per il recupero di dati da feed di dati off-chain, la formattazione dei dati in formati leggibili da blockchain, il calcolo off-chain e le garanzie di uptime che forniscono come operatori. Affinché uno smart contract su reti come Ethereum utilizzi un nodo ChainLink, dovrà pagare l'operatore del nodo ChainLink scelto utilizzando i token LINK, con i prezzi fissati dall'operatore del nodo in base alla domanda per la risorsa off-chain fornita da ChainLink e la fornitura di altre risorse simili. Il token LINK è un token ERC20, con la funzionalità aggiuntiva di "transfer and call" ERC223 di trasferimento (indirizzo, uint256, byte), che consente ai token di essere ricevuti ed elaborati dai contratti all'interno di una singola transazione.

Capitolo 6

Off-Chain Reporting

Il protocollo di Off-Chain Reporting (OCR) presentato in [3] rappresenta un miglioramento della decentralizzazione e della scalabilità delle reti Chainlink. Come presentato in precedenza, periodicamente, viene effettuata off-chain l'aggregazione delle risposte di una rete di oracoli in un unico report, utilizzando l'OCR, e trasmesso quest'ultimo a uno smart contract nella blockchain. Il contratto verifica la validità del report, paga ogni oracolo che ha contribuito con un'osservazione alla redazione del report e pubblica la mediana dei valori riportati ai contratti di consumo sulla catena. Questo comporta una maggiore efficienza dei costi, in quanto viene trasmessa una singola transazione aggregata con un notevole risparmio di gas, ma anche comitati più ampi di nodi (maggiore scalabilità) e una maggiore affidabilità durante i periodi di estrema congestione della rete blockchain. Ne consegue una maggiore decentralizzazione delle reti di oracoli, maggiore accuratezza, disponibilità e antimanomissione per gli utenti di Chainlink e una riduzione del carico di lavoro on-chain. Viene utilizzata una rete peer-to-peer sicura per la comunicazione tra i vari nodi *aggregators* e un algoritmo di consenso alleggerito in cui ogni nodo riporta la propria osservazione dei dati e la firma. Queste firme sono poi verificate nella catena.

Ciò che si vuole raggiungere con questo protocollo è resilienza a diversi tipi di fallimento, semplicità, basse tasse per le transazioni e una bassa latenza, ovvero si vuole ridurre al minimo il tempo che intercorre tra l'avvio del protocollo di firma e l'inclusione della transazione risultante nella blockchain da parte dello smart contract, in modo da avere dati aggiornati e più precisi, condizione fondamentale ad esempio per quelle applicazioni degli smart contract più sensibili ai piccoli movimenti di prezzo.
Il modo in cui il protocollo funziona è il seguente e si ripete per ogni *epoca*:

1. I nodi eleggono un nuovo nodo leader che guida il resto del protocollo per quella determinata epoca.

2. Ogni oracolo fa partire un timer di durata massima in cui il leader deve produrre un report valido da inviare. In tal modo vengono controllate le performance del leader. Se tale nodo non riesce a concludere il suo lavoro entro il tempo stabilito, inizia una nuova epoca con un leader diverso.

3. Una volta eletto, il leader chiede ai nodi di fornire osservazioni aggiornate e firmate che aggrega in un rapporto. Questo report viene inviato ai follower per verificarne la validità. Se il leader riceve una copia firmata del report da un quorum di nodi, allora tale rapporto è stato approvato. Di conseguenza, il leader assembla un report finale con le firme del quorum e lo trasmette a tutti i follower.

4. I nodi cercano di trasmettere il rapporto finale all'*aggregation contract* secondo un programma randomizzato. L'aggregatore verifica che un quorum di nodi abbia firmato il report ed

espone il valore mediano ai consumatori come risposta, insieme a un timestamp del blocco e un ID del round.

5. Tutti i nodi controllano la blockchain per il report finale al fine di eliminare qualsiasi possibile punto di errore durante la trasmissione. Nel caso in cui il nodo designato non riesca a confermare la sua trasmissione entro un determinato periodo di tempo, entra in funzione un protocollo che consente agli altri nodi di trasmettere il report finale finché uno di essi non viene confermato.

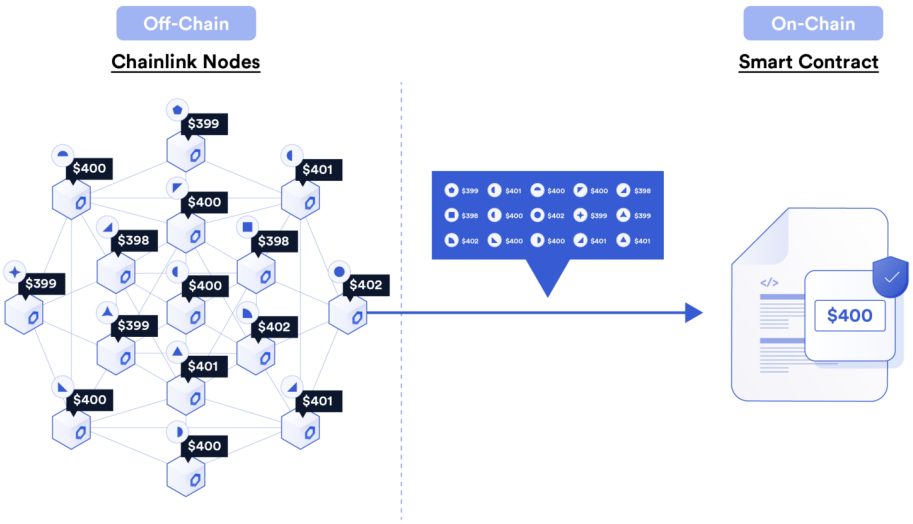

Figura 6.1. Chainlink off-chain reporting

Grazie all'introduzione dell'OCR, si è verificato un notevole aumento della quantità di dati del mondo reale disponibili per le applicazioni degli smart contracts. Ciò comporta un'ulteriore innovazione in molti settori, come la DeFi, le assicurazioni decentralizzate e i giochi basati su blockchain. Gli sviluppatori di contratti intelligenti potranno utilizzare queste nuove fonti di dati per creare una vasta gamma di prodotti e mercati finanziari. Ad esempio, in ambito DeFi è possibile aumentare sicurezza e affidabilità grazie a una maggiore precisione della valutazione del rischio. Le assicurazioni decentralizzate possono anch'esse beneficiare di tale precisione della valutazione del rischio al fine di offrire polizze personalizzate. Mentre i giochi basati su blockchain possono utilizzare i dati del mondo reale per creare esperienze di gioco più realistiche e coinvolgenti, come ad esempio giochi di ruolo basati sulla posizione geografica o giochi di simulazione basati sulle condizioni meteorologiche in tempo reale.

Capitolo 7

Conclusione

Riassumendo, in questa tesina è stata introdotta ChainLink, una rete di oracoli decentralizzata per gli smart contracts, la quale permette di interagire in modo sicuro con risorse esterne alla blockchain.
In particolare, è stata inizialmente delineata l'architettura di ChainLink, descrivendo sia i componenti on-chain che off-chain. Dopo aver definito la sicurezza nell'ambito degli oracoli, è stato prima presentato l'approccio a più livelli (multistrato) di ChainLink alla decentralizzazione ed in seguito è stato proposto un nuovo protocollo con nuove funzionalità come la protezione, con alcuni protocolli aggiuntivi, contro il freeloading. Oltre a tali protocolli, è stato citato come ChainLink può sfruttare i progressi tecnologici e infrastrutturali per migliorare la qualità dei dati forniti alle reti blockchain, ovvero utilizzando hardware affidabili e la firma digitale dei dati da parte delle fonti. Infine, è stato descritto il funzionamento dell'off-chain reporting, che ha comportato il vantaggio di ridurre il carico sulla blockchain e migliorare l'efficienza e la scalabilità.

I principi fondamentali che guidano lo sviluppo di ChainLink sono i seguenti:

- **Decentralizzazione per sistemi aperti e sicuri**. Il decentramento non è solo il fondamento delle proprietà a prova di manomissione delle blockchain, ma anche la base della loro natura senza autorizzazione. Continuando a creare sistemi decentralizzati, l'obiettivo a cui si mira è quello di consentire ulteriormente lo sviluppo senza autorizzazione all'interno dell'ecosistema. La decentralizzazione è infatti una componente cruciale per un ecosistema prospero a livello globale con sostenibilità a lungo termine.

- **Modularità per un design di sistema semplice e flessibile**. È possibile ragionare facilmente su componenti semplici e quindi combinarli in modo sicuro in sistemi più grandi. La modularità consente l'aggiornamento dei sistemi e facilita anche il decentramento. La piattaforma Chainlink si impegna a garantire che non ci sia una dipendenza eccessiva da un numero limitato di soggetti e che sia possibile utilizzare diverse implementazioni concorrenti della tecnologia.

- **Open source per sistemi estensibili e sicuri**. ChainLink è reso possibile grazie al supporto di molti progetti open source. Inoltre, la piattaforma si impegna a collaborare costantemente con sviluppatori, accademici ed esperti di sicurezza per garantire solidità e sicurezza. La piattaforma incoraggia test, audit e prove formali di sicurezza per garantire che Chainlink sia sempre all'avanguardia e in grado di soddisfare le esigenze in continua evoluzione del settore blockchain.

Forti di questi principi, gli sviluppatori di Chainlink cercano dunque di estendere la portata e l'impatto delle blockchain e degli smart contracts, rendendo gli oracoli un porto sicuro dell'ecosistema delle criptovalute.

Bibliografia

[1] S. Ellis, A. Juelsy, S. Nazarov, *ChainLink: A Decentralized Oracle Network*, September 4, 2017, [Online] https://link.smartcontract.com/whitepaper

[2] H. Al-Breiki, M. Habib Ur Rehman, K. Salah, D. Svetinovic, *Trustworthy Blockchain Oracles: Review, Comparison, and Open Research Challenges*, May 19, 2020, [Online] https://www.researchgate.net/publication/

[3] *Chainlink 2.0: Next Steps in the Evolution of Decentralized Oracle Networks*, April 15, 2021, [Online] https://research.chain.link/whitepaper-v2.pdf

[4] *Cos'è Chainlink (LINK) e come funziona*, [Online] https://cryptofacili.com/chainlink/

[5] *Chainlink: cos'è e come funziona, LINK crypto*, [Online] https://www.webeconomia.it/chainlink/#Il_Token_LINK

[6] *Chainlink: Guida definitiva su LINK Coin*, [Online] https://www.criptovaluta.it/chainlink

[7] *Basic Request Model*, [Online] https://docs.chain.link/architecture-overview/architecture-request-model/

[8] *What Is a Chainlink Node Operator?*, [Online] https://blog.chain.link/what-is-a-chainlink-node-operator

[9] *External Adapters Introduction*, [Online] https://docs.chain.link/architecture-overview/off-chain-reporting?parent=gettingStarted

[10] *Off-Chain Reporting*, [Online] https://docs.chain.link/chainlink-nodes/external-adapters/external-adapters

[11] *Chainlink Achieves Major Scalability Upgrade With Mainnet Launch of Off-Chain Reporting (OCR)*, [Online] https://blog.chain.link/off-chain-reporting-live-on-mainnet/

[12] *Video introduttivo 1: Parliamo di Chainlink (LINK): che cos'é e come funziona* https://www.youtube.com/watch?v=T8tmn3t6c5g&ab_channel=cripto51

[13] *Video introduttivo 2: Chainlink (LINK): un progetto crypto utile che devi conoscere* https://www.youtube.com/watch?v=OrLsRfHoUmw

[14] *Video introduttivo 3: Chainlink (LINK) spiegato in maniera semplice (a cosa serve, tokenomics)* https://www.youtube.com/watch?v=uTJPXcBlVaQ