Alma Mater Studiorum · Università di Bologna

SCUOLA DI SCIENZE
Corso di Laurea Magistrale in Informatica

Gestione e Indicizzazione di Dati in Contesti di Smart Transportation: un Approccio basato su Registri Distribuiti

Relatore:
Chiar.mo Prof.
Stefano Ferretti

Co-Relatore:
Chiar.mo Dott.
Mirko Zichichi

Presentata da:
Federica La Piana

II Sessione
Anno accademico 2020-21

Indice

Elenco delle figure

Introduzione

Veicoli a guida autonoma, nuovi modelli di Smart Mobility e servizi sempre più personalizzati stanno trasformando il mondo automobilistico che esige sempre più in sicurezza e affidabilità.
I sistemi di trasporto intelligenti (ITS) forniscono delle risposte economiche ed ecologiche, utilizzando vari servizi quali pianificazione intelligente del percorso ottimale in termini economici e di tempo, assistenza alla guida sicura, servizi relativi al guidatore o al veicolo e fornitura di dati ambientali.
Tali servizi tendono, in maniera sempre più ricorrente, all'uso di dati crowd-sourced e crowd-sensed, provenienti dai molteplici sensori installati sui dispositivi e sui veicoli.
La loro realizzazione, però, comporta il superamento di alcuni aspetti che devono essere considerati e che sono, fondamentalmente, legati alla raccolta, alla memorizzazione e al livello di affidabilità dei dati.
Elementi quali autenticità e verificabilità dei dati, infatti, sono fondamentali nei processi di condivisione, aggregazione e scambio di informazioni provenienti dai veicoli. Proprio su questi temi possono arrivare soluzioni dalle Distributed Ledger Technology (DLT), architetture distribuite che sfruttano protocolli di tipo Peer-to-Peer per dare vita a sistemi sicuri, anche in completa assenza di fiducia tra gli attori coinvolti.

L'obiettivo è quello di contare su transazioni trasparenti e tracciabili e facilitare l'accesso ai dati attraverso meccanismi di consenso, alta disponibilità e capacità di automatizzare e applicare processi tramite gli smart contract.
L'uso delle DLT permette, dunque, la gestione delle interazioni, ma uno degli aspetti che si presta ad ulteriori miglioramenti riguarda la ricerca delle informazioni; infatti, i dati memorizzati sui registri (ledger) sono tipicamente non strutturati e referenziati attraverso indici e indirizzi difficilmente categorizzabili; il che rende il processo di ricerca molto lento e costoso.
Nel presente lavoro viene proposto un sistema che supporta l'inserimento, l'archiviazione e il recupero dei dati crowd-sensed, basato sull'uso di una Distributed Hash Table (DHT) come strato posto sopra la DLT.
Questa soluzione fa sì che, una volta acquisiti e registrati nella DLT, i dati possano essere ricercati facilmente e rapidamente grazie alle funzionalità di ricerca offerte dalla DHT. L' aspetto interessante di quest'ultima è la particolare topologia a ipercubo che consente di raggiungere in modo efficiente gli oggetti che corrispondono a uno specifico set di parole-chiave permettendo, dunque, la costruzione di query complesse.
Relativamente alla DLT, nel presente lavoro viene preso in considerazione IOTA, il cui ledger presenta una struttura che lo rende particolarmente adatto ad applicazioni dell'Internet of Things (IoT).
Il caso d'uso per il quale tale sistema è pensato è quello della rilevazione di insidie stradali e presuppone la partecipazione, il coinvolgimento e l'interazione tra i cittadini e le pubbliche amministrazioni. Tuttavia, il sistema è potenzialmente applicabile ad altri scenari di Smart Transportation.
Il presente lavoro è organizzato come segue: la sezione II fornisce un back-

ground sulle tecnologie utilizzate; la sezione III presenta una descrizione dell'architettura del sistema; nella sezione IV viene descritto il caso d'uso e mostrata la simulazione del sistema con alcuni risultati che sono discussi nella V. Infine, nella sezione VI vengono fornite alcune osservazioni conclusive.

Capitolo 1

Background

1.1 Intelligent Transportation Systems

La saturazione delle infrastrutture di trasporto, dovuta al crescente numero di veicoli in circolazione, influisce pesantemente sulla nostra vita, soprattutto, nelle aree urbane dove l'esigenza, sempre più crescente, di muoversi rapidamente, la congestione del traffico, gli incidenti, i ritardi nei trasporti e le emissioni inquinanti dei veicoli fanno della mobilità una preoccupazione chiave [1].
Le difficoltà relative a questo aspetto della vita quotidiana stanno motivando, sempre più, la comunità di ricerca a concentrare l'attenzione sull'area dei Sistemi di Trasporto Intelligenti (ITS), al fine di trovare soluzioni capaci di contenere o risolvere del tutto alcune delle problematiche menzionate.
Tali sistemi comprendono un range di strumenti per la gestione delle reti di trasporto e servizi per i viaggiatori; essi dipendono dai risultati di attività di ricerca distribuite in molti ambiti quali, ad esempio, elettronica, robotica e sistemi informativi.

Gli ITS mirano a raggiungere l'efficienza e la dinamicità delle infrastrutture e ad incrementare la sicurezza e la comodità, fornendo agli utenti informazioni in tempo reale relative, per esempio, alla convenienza dei percorsi, alle condizioni del traffico, ai parcheggi e alla disponibilità dei posti sui mezzi di trasporto pubblico.

Un concetto emergente, che si ritiene possa aiutare nella concretizzazione degli ITS, è l'Internet of Vehicles (IoV). Tale paradigma, guidato da veicoli intelligenti, IoT e tecniche AI, permette il collegamento tra veicoli, persone e infrastrutture. Oltre ai veicoli intelligenti, l'ecosistema IoV negli ITS include anche Unità a Bordo Strada (RSU), piattaforme cloud e sistemi di comunicazione veicolare. Relativamente a questi ultimi, si parla spesso di *vehicle to everything (V2X)* [2] per indicare un paradigma che permette le comunicazioni da un veicolo a qualsiasi nodo (cioè, veicolo, infrastruttura, pedone, ecc.), e viceversa.

Naturalmente, la maggior parte degli scenari di IoV sono in tempo reale e implicano la generazione e lo scambio di grandi quantità di dati. Inoltre, in questo contesto di elevata connettività, le possibilità di attacco da parte di entità maligne sono incrementante e risulta improbabile che molte delle tecniche classiche siano adatte ed efficaci.

Le Distributed Ledger Technologies hanno il potenziale per fornire un numero sostanziale di soluzioni innovative per la maggior parte degli scenari di applicazione IoV, migliorando la sicurezza, la privacy, la fiducia e le prestazioni dei sistemi [3].

1.2 DLT e Smart Contract

Il termine Distributed Ledger Technology (DLT) [4] fa riferimento ad una famiglia di sistemi basati su un registro distribuito che può essere letto e modificato in maniera indipendente dai nodi nella rete.
Le caratteristiche di cui queste architetture decentralizzate godono le rendono una scelta ideale per scenari in cui più parti, in completa assenza di fiducia, concorrono alla gestione di dati condivisi e competono per l'accesso a tali dati. Tale scenario è tipico dei sistemi di trasporto intelligenti che sfruttano i dati rilevati da fonti multiple quali veicoli e infrastrutture.
Le proprietà di robustezza e sicurezza, per cui tali sistemi sono noti, sono garantite dall'utilizzo di tecniche crittografiche per l'archiviazione dei dati e dalla replica delle informazioni su numerosi nodi che operano come una rete Peer-to-Peer. Un'altra caratteristica particolarmente interessante è quella di immutabilità; in altre parole, una volta inseriti nel registro, i dati non possono più essere modificati o eliminati. In assenza di un'entità centrale alla quale fare riferimento, il corretto funzionamento della rete è raggiunto attraverso algoritmi di consenso che consentono ai vari nodi della rete di raggiungere un accordo circa la versione del registro da mantenere.
Meccanismo di consenso, tipologia di rete e struttura del registro rappresentano le tre fondamentali caratteristiche che distinguono i vari sistemi di Distributed Ledger.

1.2.1 Tipologia di rete

Sulla base della tipologia di rete si distingue tra sistemi:

- **Permissionless**: si tratta di sistemi concepiti per non essere controllati dal momento che permettono a chiunque, senza permesso appunto, di contribuire all'aggiornamento dei dati sul registro. Chiunque, in qualità di partecipante può diventare un validatore e può disporre di tutte le copie immutabili di ogni operazione. IOTA e Ethereum rientrano in questa categoria.

- **Permissioned**: permettono di definire speciali regole per l'accesso e la visibilità dei dati introducendo, dunque, un concetto di governance limitata a pochi nodi presenti. Il sistema di approvazione delle transazioni, infatti, non è vincolato alla maggioranza dei partecipanti, bensì, a un numero limitato di attori. Hyperledger Fabric [5] è un esempio di DLT permissioned.

1.2.2 Smart Contracts

Un concetto strettamente connesso alle DLT è quello degli *Smart contract* [6]. Nel senso tradizionale, un contratto è un accordo tra due o più parti in cui ciascuna ottiene un beneficio in cambio di qualcos'altro. Questo implica la necessità delle parti di fidarsi l'una dell'altra per l'assolvimento degli obblighi. Lo smart contract è, di fatto, un programma basato su tecnologia Blockchain che si rifà al concetto di contratto tradizionale, ma generalmente non può essere equiparato ad un contratto legale.
Lo scopo di uno smart contract, infatti, è quello di implementare un processo di accordo tra due parti direttamente sulla blockchain. Tuttavia, la tipologia di accordo è limitata dalla tecnologia che esegue lo smart contract.

Gli esempi di smart contract più comuni sono associati alle piattaforme blockchain come Ethereum [7] e Bitcoin [8].
Lo scopo degli smart contract è quello di proporre un paradigma di "fiducia senza fiducia" [9] in cui, cioè, la fiducia è spostata da un intermediario umano al protocollo stesso. Essendo basati sulla blockchain, gli smart contract, infatti, ne ereditano alcune interessanti proprietà:

- **Immutabiltà**: dopo la creazione, uno smart contract non può più essere modificato. Nessuno può cambiare il codice del contratto, nemmeno il creatore del contratto stesso.

- **Decentralizzazione**: l'unione tra la blockchain e la tecnologia degli smart contract consente, di fatto, di rimuovere qualsiasi dipendenza delle parti coinvolte da sistemi centrali. Infatti, l'esecuzione degli smart contract avviene in maniera distribuita, ovvero, tutti i partecipanti alla rete ricevono gli stessi input ed eseguono una computazione che conduce agli stessi output. Questo principio si basa essenzialmente sul presupposto che la maggioranza dei partecipanti sia onesta.

Un altro elemento che contribuisce a rendere gli ecosistemi DLT ancora più appetibili per aziende e start-up, che possono sfruttarne le potenzialità per fini commerciali, è rappresentato dai *token*.
Questi ultimi sono gestiti dagli smart contract che ne definiscono le caratteristiche, per esempio, la quantità in circolazione, le regole di accesso, chi è abilitato a trasferirli ecc. I token possono, potenzialmente, rappresentare qualsiasi cosa, per esempio il possesso di asset reali o digitali, diritti di accesso a beni, possibilità di svolgere azioni ecc.

Questa possibilità ha portato alla coniazione di un nuovo termine "tokenizzazione" per indicare il fenomeno di "trasposizione" su blockchain di oggetti, diritti e valori.
Considerato l'ampio utilizzo dei token di Ethereum in diverse applicazioni decentralizzate, sono stati proposti diversi modelli. Lo standard ERC20 [10], per esempio, è la scelta più comune per le criptovalute. D'altra parte, il token ERC721 [11] è di utilità non fungibile, solitamente implementato per rappresentare qualcosa di unico.

1.3 IOTA

IOTA [12] è un progetto avviato nel 2015 da David Sønstebø, Sergey Ivancheglo, Dominik Schiener e Dr. Serguei Popov, pensato per il campo dell'IoT e nato dalla necessità di superare alcuni limiti intrinseci delle classiche architetture blockchain. IOTA può essere considerato un ecosistema formato da due componenti principali, un registro distribuito permissionless e l'omonima criptovaluta.

1.3.1 Proprietà

Decentralizzazione

Nelle classiche architetture blockchain il meccanismo di conferma delle transazioni si basa sul mining, ossia, sull'intervento dei *miners* che, incentivati da un premio e dalle commissioni che il mittente paga al momento della creazione della transazione, eseguono la Proof of Work (PoW), necessaria per la creazione dei nuovi blocchi. Negli ultimi anni, tuttavia, si è assistito ad una vera e

propria corsa al mining, concretizzatasi nella creazione di enormi agglomerati di miner detti *mining pools* che, collaborando tra di loro, minacciano la decentralizzazione della rete, disincentivando i piccoli miner ad eseguire la PoW.
IOTA argina tale problema, eliminando la necessità di avere miner esterni per confermare le transazioni. Come si vedrà in seguito, il compito di validazione spetta, infatti, a chiunque voglia eseguire una transazione.

Scalabilità

Nelle classiche blockchain l'inserimento a velocità costante di un blocco, mediamente uno ogni 10 minuti, rappresenta di fatto un collo di bottiglia per le prestazioni della rete.
Al contrario, il protocollo IOTA prevede che, per ogni transazione immagazzinata nel registro, questa ne approvi altre due già presenti. Questo meccanismo lo rende, almeno in teoria, infinitamente scalabile. Infatti, in base agli studi effettuati dalla IOTA Foundation [13] [14], più nodi partecipano alla rete, più velocemente le transazioni verranno approvate, con conseguenti vantaggi in termini di efficienza e sicurezza.

Zero Fee

In IOTA, l'onere della PoW, il cui livello di difficoltà dipende dalla volontà del mittente, spetta a chiunque intenda effettuare una transazione. Questo permette di eliminare la figura del miner e, di conseguenza, la necessità di pagare delle commissioni (fee) per le transazioni eseguite.
Il conseguente vantaggio è la possibilità di effettuare micropagamenti, ossia,

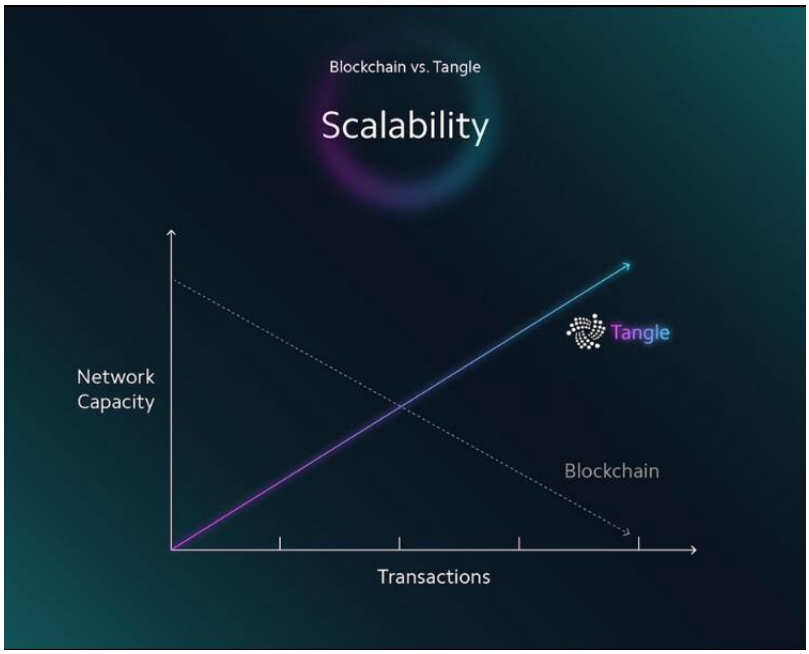

Figura 1.1: Scalabilità di IOTA [15]

transazioni contenenti pochissima valuta o, addirittura, prive; il che rende IOTA particolarmente adatto all'IoT.

1.3.2 Il Tangle

A rendere IOTA sostanzialmente diverso dalla classica blockchain è l'elemento fondamentale sul quale si basa, il Tangle [16], che è la struttura dati utilizzata per il registro distribuito. Esso si basa su un *Grafo Aciclico Diretto (DAG)*, composto da vertici e da archi. La rappresentazione grafica è mostrata nella figura 1.2

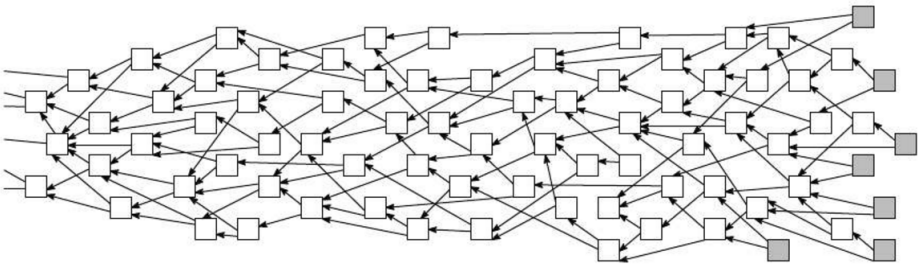

Figura 1.2: Rappresentazione grafica del Tangle di IOTA [17]

1.3.3 Componenti

I vertici, detti anche nodi, rappresentano le singole transazioni, mentre gli archi, che hanno una direzione, indicano i collegamenti tra due nodi.
Come accennato precedentemente, ogni nuova transazione inserita nel Tangle contribuisce all'approvazione di due transazioni precedenti, scelte in maniera pseudo-casuale tra quelle non ancora referenziate (*tips*). L'approvazione può avvenire in maniera diretta, se esiste un arco che collega le due transazioni, o indiretta, se esiste un percorso che consente di arrivare da una transazione all'altra.
La prima transazione avvenuta nel Tangle è definita *genesis transaction* ed è l'unica che ha generato *token* all'interno di IOTA.
I nodi che intendono emettere transazioni sul Tangle devono:

- Scegliere due *tip* da approvare, secondo un particolare algoritmo noto come *Markov Chain Monte Carlo (MCMC)*;
- Accertarsi che le due transazioni scelte non siano in conflitto e non approvare quelle che lo sono;

- Svolgere una *Proof-of-Work* simile a quella di Bitcoin, ma molto meno dispendiosa;

Le transazioni all'interno del Tangle vengono caratterizzate da un peso (*weight*) che è proporzionale alla quantità di Proof-of-Work eseguita durante l'operazione di inserimento e che, in tal senso, determina l'importanza della transazione stessa.
Il *cumulative weight* è, invece, definito come il peso di una transazione più la somma dei pesi delle transazioni che la approvano direttamente o indirettamente. La figura 1.3 mostra una porzione del Tangle in cui per ciascuna transazione vengono mostrati i valori dei parametri *weight* (in basso a destra) e *cumulative weight* (in alto a sinistra).
È da notare, inoltre, come i *cumulative weight* delle transazioni cambino in seguito all'arrivo della transazione X che approva due *tip*.

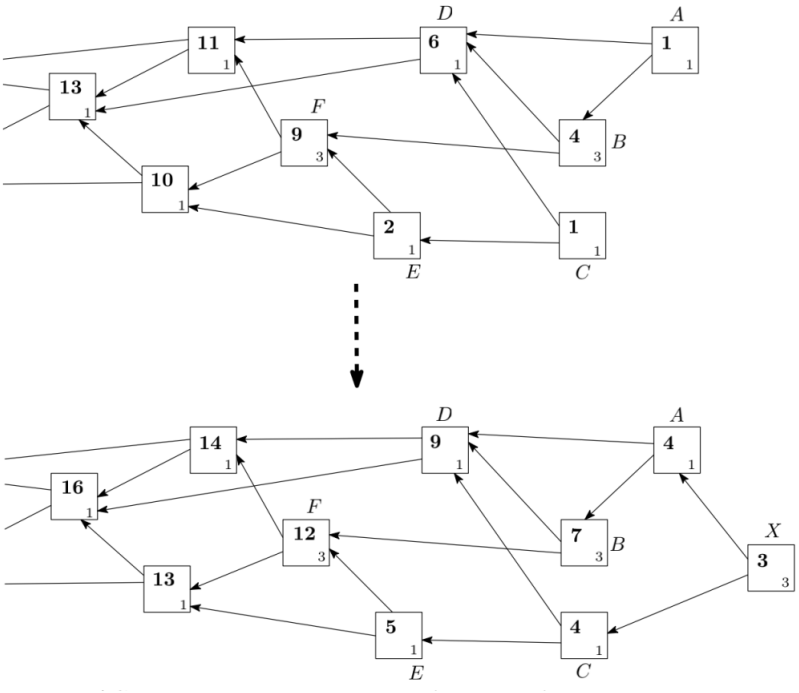

Figura 1.3: Weight e Cumulative weight prima e dopo l'arrivo di una nuova transazione nel Tangle [17]

L'altezza (*Height*) indica la lunghezza del percorso più lungo fino alla transazione di genesi. Per esempio, nella figura 1.3 l'altezza della transazione G è 1, mentre D ha un'altezza pari a 3.

La profondità (*Depth*) è, invece, il numero di transazioni che intercorrono nel più lungo percorso inverso fino ad una tip.

Nel caso sopra raffigurato G ha una profondità di 4 transazioni fino alla tip A, che è la più lontana.

1.3.4 Nodi della DLT

I nodi, centrali nell'architettura di IOTA, sono programmi software installati sulle macchine che partecipano alla rete. Essi, attraverso operazioni di

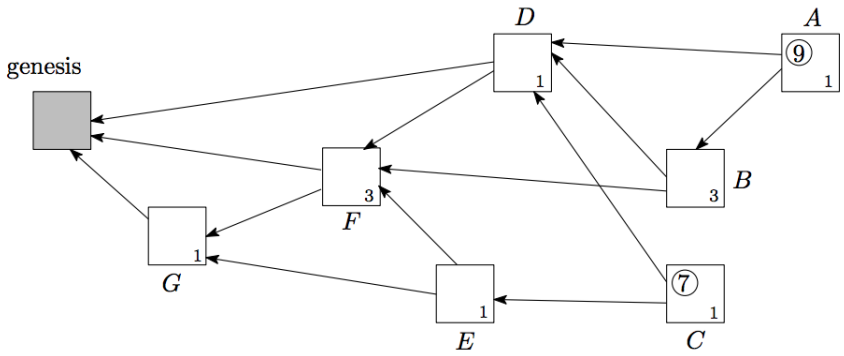

Figura 1.4: Depth e Height delle transazioni nel Tangle [17]

lettura e scrittura nel registro, rendono possibile accedere al Tangle. Esistono diverse tipologie di nodo:

- **Full Node**: si tratta del nodo più completo in termini di funzionalità offerte per l'interazione con il Tangle e con il resto della rete. Questa tipologia di nodo memorizza l'intero Tangle ed è in costante comunicazione con i propri vicini (*neighbors*).

  L'IRI (*IOTA Reference Implementation*) è il software Java che implementa le funzionalità di un Full Node, esponendo una serie di API che permettono la gestione dei vicini e delle transazioni nel Tangle.

- **Light Node**: questo tipo di nodo può essere visto come una versione limitata del Full Node in quanto non mantiene una copia locale del Tangle; dunque, necessita di fare riferimento ad esso per accedervi. È pensato per dispositivi e macchine con scarse capacità computazionali.

- **Coordinator**: il Coordinator è un nodo speciale, controllato dalla IOTA Foundation, pensato per proteggere la rete, non ancora sufficientemente grande, da possibili attacchi quali quello del 51% [18]. Lo scopo primario del Coordinator, infatti, è quello di "vegliare" sul Tangle emettendo,

periodicamente, una transazione definita *Milestone*, utilizzata per confermare direttamente o indirettamente le transazioni nel Tangle. Quelle verificate da una milestone non sono falsificabili in quanto approvate direttamente dal Coordinator.

L'impiego di tale nodo è, tuttavia, temporaneo; infatti, non appena la rete raggiungerà un alto valore di transazioni confermate al secondo (CTPS), il Coordinator verrà disattivato [19].

1.3.5 Transazioni

Le transazioni sono trasferimenti di dati da un indirizzo all'altro, trasmessi alla rete e inseriti nel Tangle. Quando l'oggetto del trasferimento non è rappresentato da tokens, si parla di transazioni *zero-value*. Ciascuna transazione, la cui struttura è descritta nella tabella 1.1, è composta da 2,673 trytes e può far parte di un pacchetto di transazioni, ovvero, un *bundle*.

1.3.5.1 Indirizzi

Un indirizzo appartiene ad un *seed* e rappresenta l'informazione che permette di spostare i token da un wallet ad un altro. Per *wallet* si intende lo strumento che permette all'utente di possedere e mantenere al sicuro una certa quantità di valuta, in questo caso, IOTA tokens.

Gli indirizzi rappresentano la metà pubblica di una coppia di chiavi pubblica/privata. Per inviare una transazione da un indirizzo ad un altro occorre firmarla con la chiave privata, nota solo al proprietario del seed. La parte pubblica, ovvero l'indirizzo, è condivisibile dal momento che non permette di risalire nè alla chiave privata, nè al seed.

Campo	Descrizione	Trytes
hash	Hash della transazione	81
signatureMessageFragment	Una firma o un messaggio, entrambi possono essere frammentati in un bundle	2187
address	Contiene l'address di chi invia o di chi riceve tokens	81
value	Quantità di IOTA tokens scambiata.	27
timestamp	Timestamp arbitrario	27
currentIndex	Indice della transazione nel bundle	9
lastIndex	Indice dell'ultima transazione nel bundle	9
bundle	Hash usato per raggruppare più transazioni in un bundle	9
trunkTransaction	Hash di una transazione esistente nel Tangle che la transazione corrente referenzia	81
branchTransaction	Hash di una transazione esistente nel Tangle	81
attachmentTag	Tag definito dall'utente	27
attachmentTimestamp	Timestamp dopo la terminazione della PoW	9
attachmentTimestampLowerBound	Lower Bound	9
attachmentTimestampUpperBound	Upper Bound	9
nonce	Stringa generata durante la risoluzione della PoW	27

Tabella 1.1: Anatomia di una transazione [12]

Oltre al seed dell'utente, la generazione degli indirizzi richiede altri due parametri, un *index* e un *security level*. Quest'ultimo può assumere un valore compreso tra uno e tre ed indica la difficoltà utilizzata nella generazione di chiavi e indirizzi.

$$indirizzo = seed + indice + livello\ di\ sicurezza$$

1.3.5.2 Selezione delle Tip

Una volta che la transazione è stata creata e firmata con la chiave privata, il passo successivo consiste nell'approvare due transazioni non confermate (tips). La selezione delle tip, normalmente effettuata da un Full Node in seguito ad una richiesta, avviene secondo l'algoritmo Markov Chain Monte Carlo (MCMC). Più specificatamente, il nodo seleziona una porzione del Tangle (*subtangle*), che va da una milestone ad un gruppo di tip, ed esegue due cammini casuali che restituiscono le transazioni da approvare. Durante il cammino, la scelta della successiva transazione su cui muoversi non è del tutto casuale, ma dipende dal *cumulative weight* di ognuna di esse, ovvero, il peso di una transazione più la somma dei pesi di tutte quelle che la approvano direttamente o indirettamente.

1.3.5.3 Proof-of-Work

L'ultimo passaggio richiesto per inserire una transazione sul Tangle consiste nell'eseguire la Proof-Of-Work, simile a quella richiesta in registri miner-based come Bitcoin, ma più semplice, il cui scopo è principalmente quello di prevenire gli attacchi di Spam e di Sybil [20].
La funzione di PoW prende in input l'oggetto transazione codificato in trytes e

il Minimum Weight Magnitude (la difficoltà della PoW), restituendo un numero detto *nonce*. I valori dei campi della transazione insieme al nonce vengono convertiti in trits e sottoposti ad hash, usando la funzione CURL [21].
Questo processo continua fino a quando l'hash della transazione non termina con un numero di zeri uguale o superiore al valore del Minimum Weight Magnitude (MWM); in tal caso il nonce è considerato valido.

1.3.6 MAM

Sebbene la sua natura fee-less lo renda particolarmente adatto al mondo dell'IoT, dal momento che non prevede commissioni per le transazioni, IOTA permette di pubblicare un solo messaggio alla volta.
Questa limitazione si rivela problematica in contesti in cui vi è la necessità di mantenere messaggi consecutivi collegati. Si pensi al caso in cui si volesse pubblicare sul Tangle la temperatura che un sensore rileva ogni 20 secondi. In contesti come questo, per preservare l'ordine dei dati, occorrerebbe inviare le transazioni, contenenti il dato, tutte allo stesso indirizzo, il che renderebbe semplice effettuare attacchi di Spam dal momento che l'indirizzo potrebbe essere facilmente individuato da attori malintenzionati.
Il *Masked Authenticated Messaging* [22] (MAM)è un protocollo di comunicazione introdotto da IOTA nel 2017 che consente di pubblicare o consultare flussi di dati cifrati sul Tangle, pensato proprio per colmare la limitazione sopra indicata; infatti, utilizzando indirizzi diversi, permette di legare tra loro transazioni eseguite dallo stesso *seed*, appartenente all'entità che pubblica i messaggi (es. il sensore IoT).

1.3.6.1 Canali

MAM si basa sull'utilizzo di canali (*channel*) che, similmente a quanto avviene su altre piattaforme quali YouTube, consentono al proprietario (il possessore del seed) di pubblicare contenuti e ad altri utenti di iscriversi per accedere ad essi.

Il *seed*, che sancisce la proprietà del canale, deve essere conservato in maniera sicura in quanto permetterebbe a chiunque ne entri in possesso di pubblicare messaggi come se ne fosse il reale proprietario.

Il flusso di messaggi è a senso unico; infatti, ciascuno di essi ha un identificativo definito *root* e conosce l'indirizzo *root* del successivo (*nextRoot*), motivo per cui gli utenti, a conoscenza di un punto d'accesso, potranno leggere i messaggi solo da quel punto in poi.

In base al livello di visibilità e al meccanismo di accesso ai dati si distinguono 3 tipologie di canali:

- **Public**: in questa tipologia di canale la stringa *root* rappresenta sia l'indirizzo di un messaggio che la chiave di cifratura. Pertanto, chiunque la possieda, può accedere al messaggio, decodificarlo e seguire il flusso per recuperare gli elementi successivi. La modalità pubblica è tipicamente utilizzata per creare flussi di dati destinati ad essere di pubblico dominio.

- **Private**: in questa modalità l'indirizzo è rappresentato dall'hash della *root*, mentre quest'ultima funge da chiave per decifrare il messaggio. Conoscere l'indirizzo, dunque, non è sufficiente per leggere il messaggio dal momento che la root non è ricavabile dal suo hash.

La modalità privata è tipicamente utilizzata in situazioni in cui si ha necessità di limitare la condivisione di informazioni.

- **Restricted**: in questo caso l'indirizzo è l'hash della root, ma per decifrare il messaggio è necessaria una chiave di autorizzazione (SideKey). La sideKey è scelta dal proprietario che decide anche a chi fornirla.

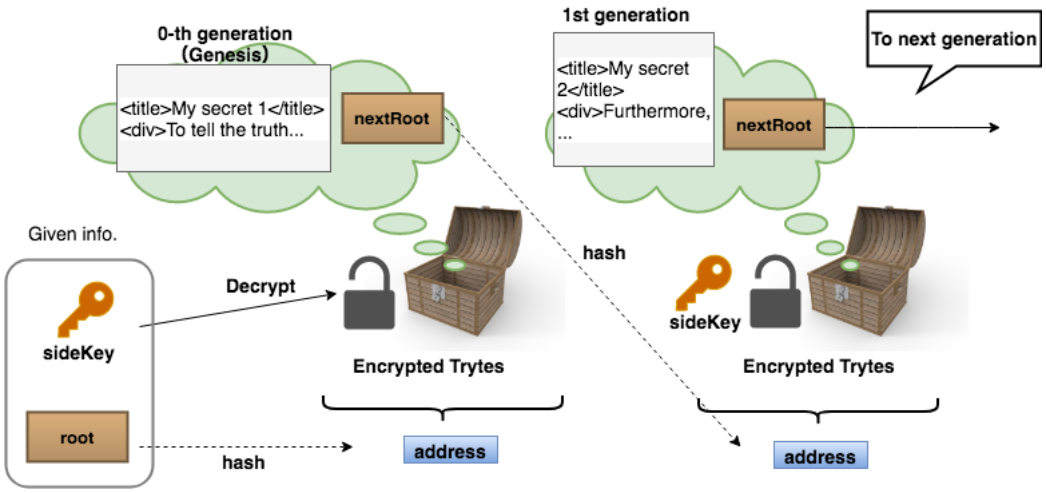

Figura 1.5: Rappresentazione grafica del canale MAM in modalità restricted [23]

1.3.6.2 Merkle Tree

Per la creazione delle root MAM utilizza un Merkle Tree a partire da alcuni indirizzi generati dal seed.

Un Merkle Tree è un albero binario in cui le foglie sono rappresentate dall'hash di un blocco di dati e ogni nodo non foglia è creato a partire dalla combinazione dei sui nodi figli. Il risultato finale è, dunque, un singolo hash, ovvero, la root. La funzione di generazione del Merkle Tree richiede in input due parametri, *Start* e *Size*. Il primo indica l'indice (*index*) di partenza per la generazione

degli indirizzi relativi a un seed, il secondo riguarda il numero di indirizzi da generare e, dunque, la dimensione dell'albero.

La creazione del Merkle Tree è un processo bottom-up strutturato in fasi:

1. Utilizzando il *seed*, l'*index* e un *security level* viene generata la chiave privata (*side key*) con la quale si eseguirà la cifratura dei messaggi;
2. L'hashing della chiave permette di ricavare l'indirizzo al quale verranno inviati i messaggi;
3. Un'ulteriore operazione di hashing consente di ricavare il valore dei nodi foglia (leafs);
4. Eseguendo l'hash delle foglie, a due a due, si ottiene il valore finale, ovvero la *root*;

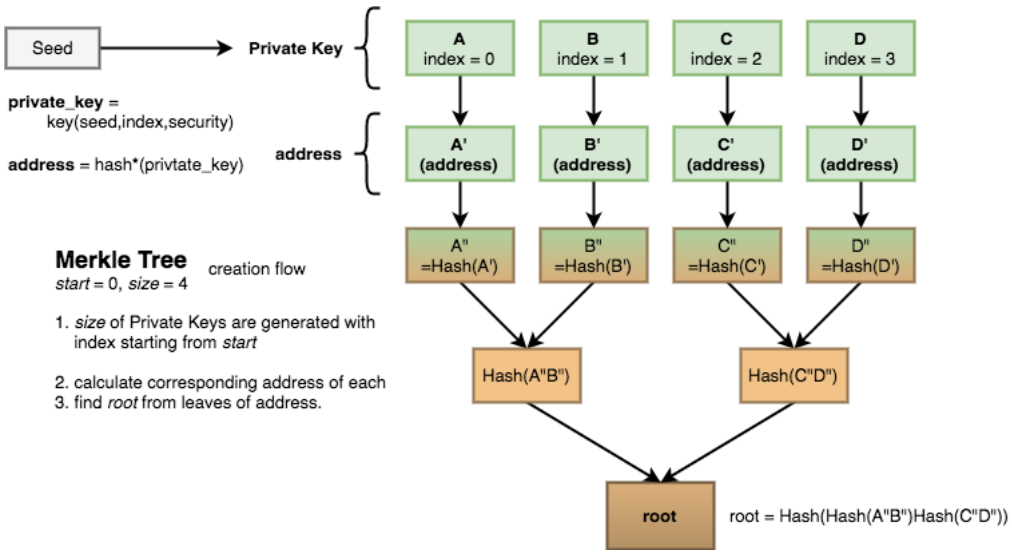

Figura 1.6: Creazione del Merkle Tree [23]

1.3.6.3 Messaggi

Come detto precedentemente, il meccanismo di pubblicazione dei messaggi di MAM prevede l'utilizzo di indirizzi sempre diversi in cui, però, sono conte-

nute informazioni che permettono all'utente di seguire il flusso di transazioni. Un pacchetto MAM è composto da almeno 3 messaggi ed è strutturato in due sezioni, sezione MAM e sezione della firma.

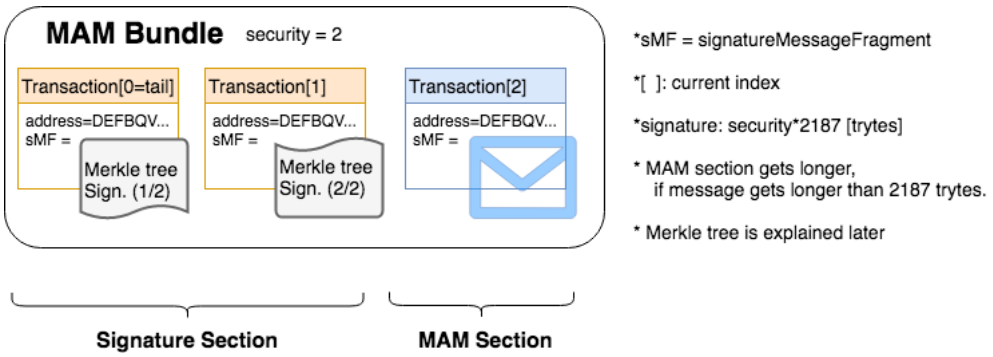

Figura 1.7: Struttura del MAM Bundle [23]

MAM section

In questa sezione sono presenti i valori di *message*, *nextRoot*, *branch_index* e *Siblings*, criptati con la root (nella modalità pubblica) o con la sideKey (nella modalità Restricted).

- **Message**: contiene il messaggio vero e proprio convertito in trytes.
- **NextRoot**: rappresenta il parametro che consente di collegare una transazione alla successiva. Per pubblicare un messaggio, infatti, sarà necessario generare due Merkle Tree in modo tale da ottenere due root, quella per il messaggio corrente e quella per il messaggio successivo.
- **Branch_index**: all'interno di un singolo Merkle Tree il campo *branch_index* indica l'indice di ogni singola foglia. Nell'esempio in figura 1.6, un albero con $start = 0$ e $count = 4$ avrà $branch\_index = 0, 1, 2, 3$.

- **Siblings**: a partire dall'address di branchIndex, siblings contiene un insieme di hash complementari che, combinati insieme all'*address*, danno origine alle root. In riferimento alla figura 1.6, preso un branchIndex 0, i siblings per ottenere la root sono B" e Hash(C"D").

Signature section

Tale sezione contiene la firma apposta al bundle, necessaria per verificarne la validità.

Dal processo di validazione, effettuato al momento del recupero del messaggio (*MAM fetch*), si ricava un *address*, il quale rappresenta una delle foglie del Merkle tree, esattamente quello dove index = *branch_Index*.

Combinando questo valore con i siblings all'interno della transazione, si ottiene una root (*temp_root*). Il messaggio è considerato valido se i valori della root e della *temp_root* sono uguali.

1.4 DHT

Una Distributed Hash Table (DHT) è un sistema di archiviazione decentralizzato che fornisce schemi di ricerca e memorizzazione simili a quelli di una tabella hash, permettendo l'archiviazione di dati sottoforma di coppie chiave-valore. Consiste in una rete peer-to-peer di nodi che si occupa delle gestione dei dati e di un meccanismo di routing che permette la ricerca di oggetti nella rete.

In questo lavoro viene presa in considerazione una DHT strutturata con caratteristiche mostrate di seguito. L'associazione degli oggetti ai nodi della DHT è ottenuta mediante una funzione hash, ovvero, una funzione unidirezionale

che mappa qualsiasi oggetto in una stringa binaria di n-bit. Quest'ultima, dunque, rappresenta la chiave dell'oggetto e consente una corretta e bilanciata distribuzione delle informazioni tra i nodi della DHT.

I nodi sono, a loro volta, identificati attraverso un ID a n-bit che risiede nello stesso spazio ID usato per identificare i contenuti. Ciascuno nodo sarà, dunque, responsabile della gestione di una porzione dello spazio degli ID corrispondente ad un sottoinsieme dei dati. Una rete DHT può essere modellata con un grafo *G=(V,E)*, dove V rappresenta l'insieme di nodi ed E l'insieme di archi che li collegano. L' insieme dei nodi costituisce l'overlay network che viene organizzato in modo strutturato dando, così, forma alla topologia della rete.

Una specifica topologia è quella a ipercubo [24] che si rivela particolarmente adatta per l'esecuzione di query, basate su keyword.

1.4.1 Struttura a ipercubo

Un ipercubo a r dimensioni $H_r(V,E)$ è un grafo composto da 2^r nodi. Ciascuno nodo $u \in V$ è rappresentato da un vettore binario di dimensione r. Con *u[i],* $0 \leq i \leq r-1$, si intende l' i-esimo bit del nodo u a partire da destra. Per ciascuna coppia di nodi *u,v* in V esiste una arco *(u,v)* in E se e solo se i due nodi differiscono per un solo bit.

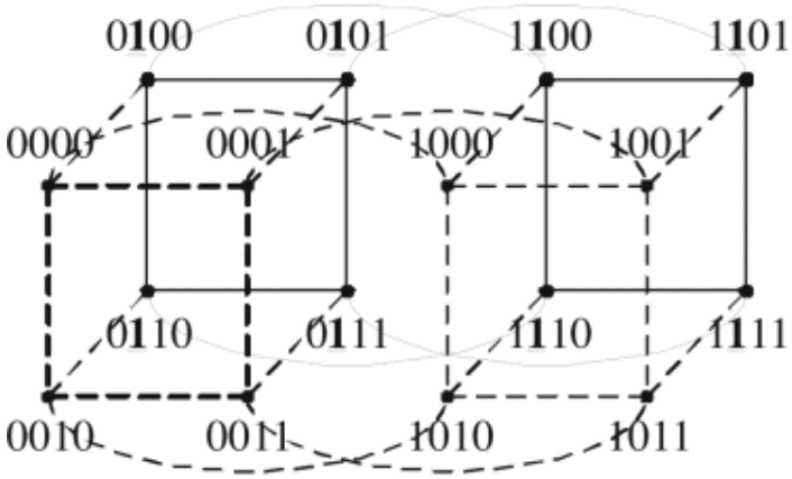

Figura 1.8: Rappresentazione grafica di un ipercubo [24]

Come esempio, in figura 1.8 è mostrato un ipercubo a quattro dimensioni. Ogni nodo è, dunque, rappresentato da una stringa a 4 bit. I nodi u=1010 e v=1011 sono da considerare vicini, in quanto differiscono tra loro di un solo bit.

Inoltre, per ogni nodo $u \in V$ si definiscono gli insiemi *One(u)* e *Zero(u)* che rappresentano, rispettivamente, le posizioni dei bit con valore 1 e le posizioni dei bit con valore 0 del nodo u. Formalmente, $One(u) = \{i \mid u[i] = 1,\ 0 \leq i \leq r-1\}$ e $Zero(u) = \{\ i \mid u[i] = 0,\ 0 \leq i \leq r-1\ \}$.

Con riferimento alla figura 1.9, il nodo u=1010 ha un valore di One(u)={1,3} e Zero(u)={0,2}.

1.4.1.1 SBT e Distanza di Hamming

Per comprendere meglio la ricerca di oggetti all'interno della DHT è necessario definire i concetti di sub-ipercubo e distanza di Hamming.

Il sub-ipercubo $H_r(u)$ di un nodo u è un sottografo G=(U,F) dell'ipercubo H_r in cui ciascun nodo $w \in V$ si trova in U se e solo se w contiene u e ogni arco $e \in E$ si trova in F se e solo se i suoi end points si trovano in U.

In figura 1.9 è riportato un esempio del sub-ipercubo indotto dal nodo 0100.

Al fine di capire quanto due nodi della rete distano l'uno dall'altro, si utilizza la distanza di Hamming.

Nello specifico, date due stringhe u=$\{u_i, u_{i-1},.., u_o\}$ e v=$\{v_i, v_{i-1},.., v_o\}$, dove $0 \leq i \leq r-1$, la distanza di Hamming è il numero di posizioni i in cui $u_i \neq v_i$.

Nel caso di due stringhe binarie, la distanza di Hamming è rappresentata dal numero di bit impostati a 1 risultanti a seguito di un'operazione di XOR.

$$Hamming(u,v) = \sum_{i=0}^{r-1}(u[i] \oplus u[i])$$

A titolo di esempio, i nodi u=1010 e v= 1011 sono vicini l'uno all'altro, dal momento che presentano una distanza di Hamming pari a 1.

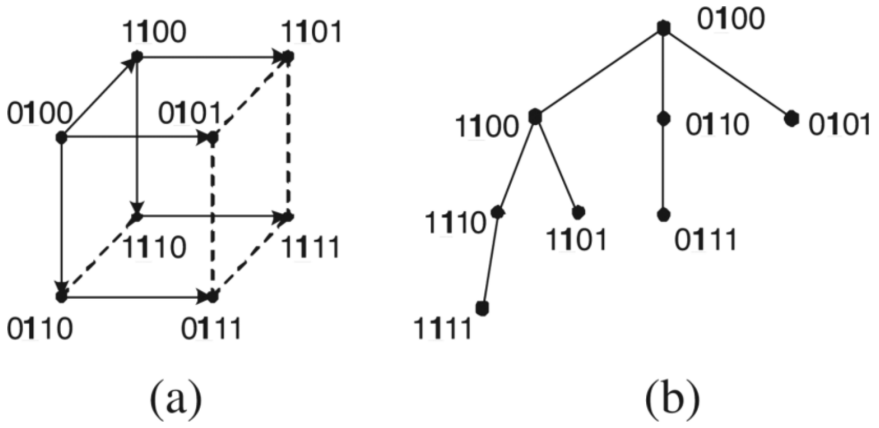

Figura 1.9: Rappresentazione grafica dell'ipercubo e relativo Spanning Binomial Tree [24]

Sulla base del sub-ipercubo $H_r(u)$ si ricava il corrispondente Spanning Bi-

nomial Tree *SBT(u)*, di fondamentale importanza per la ricerca di oggetti all'interno della rete DHT.

Un SBT, che contiene esattamente lo stesso numero di nodi presenti nel relativo sub-ipercubo, ha come radice u e come nodi a livello 1 tutti quelli che contengono u e si discostano da esso solo di un bit. Tale regola si ripete per ogni livello. Pertanto, all'interno di un generico *SBT(u)* il nodo v, che si trova al j-esimo livello dell'albero, avrà esattamente j bit di differenza dal nodo radice. Questa proprietà è veicolare nel meccanismo di ricerca basato su keyword.

1.4.2 DHT basata su ipercubo

Chiariti i concetti di ipercubo e DHT, il passo successivo riguarda la loro integrazione.

Per costruire un ipercubo su una rete DHT fisica è necessario un mapping tale che ogni nodo logico dell'ipercubo abbia un corrispondente nodo fisico nelle rete.

Un aspetto importante da considerare è che la dimensione dell'ipercubo, determinata dal numero di oggetti da indicizzare, può essere disaccoppiata da quella della DHT, rappresentata dal numero di nodi che partecipano al sistema.

Ciò significa che i nodi logici, ossia quelli dell'ipercubo, possono essere in numero maggiore dei nodi fisici, cioè quelli presenti nella DHT. Se si considera questo caso, si viene a creare uno scenario in cui un nodo fisico si occupa di più nodi logici. Se la situazione fosse inversa, solo una frazione di nodi fisici sarebbe responsabile dell'indicizzazione degli oggetti.

1.4.2.1 Schema di indicizzazione

Sia W l'insieme di tutte le keyword considerate nel sistema. La funzione $h : W \rightarrow \{0, 1, .., r-1\}$ mappa ogni keyword in un numero intero da 0 a r-1. Si definisce, inoltre, il mapping $F_{\mathrm{h}} : 2^W \rightarrow V$ che consente di determinare il nodo responsabile di un set di keyword.
$F_{\mathrm{h}}(\mathrm{K}) = u$ se e solo se $One(u) = \{h(w)|w \in K\ \}$. In altre parole, F_{h} è il nodo in cui i bit sono impostati a 1 nella stessa posizione ricavata con la funzione hash h in base alle keyword in k.
Dunque, per ogni possibile set di keyword c'è un unico nodo responsabile. Si noti, inoltre, che un nodo può essere responsabile di più di un set di keyword. Come esempio, si prenda in considerazione un ipercubo a 4 dimensioni (r=4) e il set di keyword W=$\{a,\ b,\ c,\ d\}$. Per semplicità, ciascuna keyword viene mappata con il numero corrispondente alla sua posizione all'interno del set.
Per esempio, per il keyword set {a,c,d} si ha *h(a)*=0, *h(c)*=2, *h(d)*=3.
Leggendo i bit partendo da destra il nodo responsabile per il set è 1101.

1.4.2.2 Ricerca

Ciascun nodo u dell'ipercubo H_{r} mantiene una index table degli oggetti nel formato chiave-valore, dove il valore è l'oggetto vero e proprio che si intende memorizzare nella rete e la chiave è data dalle keyword associate all'oggetto. Oltre alle operazioni di inserimento e cancellazione, è possibile effettuare due tipi di ricerca, *Pin Search* e *Superset Search*.

- **Pin Search**: si tratta di una ricerca precisa, mirata ad ottenere tutti e solo quegli oggetti associati ad un determinato set di keyword K, cioè, $\{o$

$\in O \mid K_o = K\}$. Su richiesta, dunque, il nodo responsabile del set di keyword specificato restituirà al richiedente tutti gli oggetti corrispondenti a K.

- **Superset Search**: simile alla precedente, ma cerca anche quegli oggetti descritti da insiemi di parole chiave che includono K, cioè, $\{o \in O \mid K_r \subseteq K\}$.

  A differenza della Pin Search, che coinvolge un singolo nodo, la ricerca Superset coinvolge anche quei nodi che non sono strettamente responsabili del set di keyword specificato.

  A livello pratico, individuato il nodo responsabile, viene calcolato il suo sub-ipercubo interrogando, dunque, anche i nodi che ne fanno parte. La ricerca nel sub-ipercubo viene effettuata esplorando l'albero SBT del nodo radice. Così facendo, all'aumentare della profondità dell'albero, ovvero del numero di livelli, la ricerca diviene sempre più specifica. Pertanto, il nodo v al j-esimo livello di *SBT(u)* avrà nel suo set j keyword in più rispetto al set K iniziale.

  Com'è facile intuire, i risultati di tale ricerca possono essere piuttosto ampi, tuttavia, è possibile stabilire una soglia c che indica la quantità massima di oggetti da restituire.

1.4.2.3 Meccanismo di routing

L'algoritmo di routing è alla base del processo di ricerca degli oggetti all'interno dell'ipercubo dal momento che garantisce l'individuazione del percorso ottimale tra due nodi.

Al momento dell'esecuzione della query viene contattato un qualsiasi nodo $v \in V$ della rete. Se il nodo in questione non è il responsabile del keyword set K, la richiesta viene inoltrata a uno dei vicini. Tale processo viene ripetuto ricorsivamente finché la query non raggiunge il nodo responsabile.

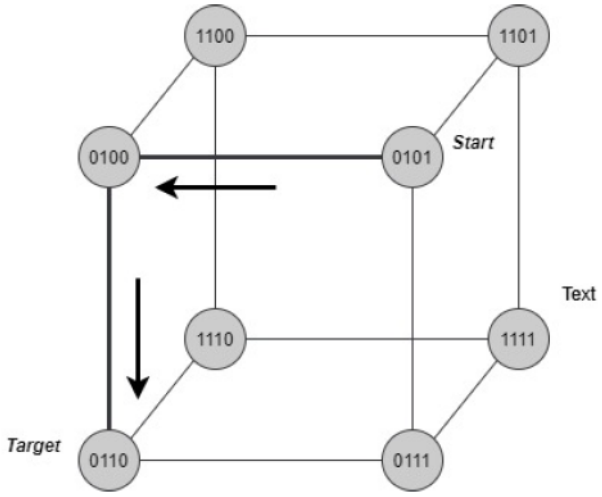

Figura 1.10: Routing ottimale tra due nodi [25]

In figura 1.10 è mostrato un esempio di routing. Si supponga che il nodo con ID=0101 sia il nodo di partenza, mentre il nodo con ID=0110 sia il target da raggiungere; dal momento che i due nodi presentano più di un bit di differenza, non sarà possibile raggiungere il nodo target con un collegamento diretto, ma il nodo di partenza dovrà inoltrare la richiesta a quel nodo che, tra i suoi vicini, è più vicino al target, ovvero quello che ha distanza di Hamming minore.
Nel caso in cui vi siano più percorsi ottimali, ossia che richiedano lo stesso numero di passi come nel caso in figura 1.10, l'algoritmo sceglie sempre il primo che incontra. Il processo di routing è descritto dettagliatamente nell'algoritmo in figura 1.11.

Algorithm 1: QueryRoutingMechanism

Input: q query, K keyword set, l limit
Data: v node string, $one(v)$, $neighbors(v)$
Result: $\{o \in O \mid K_o \supseteq K\}$

```
1  one(u) ← {h(k) | k ∈ K}
2  if one(u) ≠ one(v) ∧ From(q) = "User" then
3      w ← {n | n ∈ neighbors(v)∧
         Min(Hamming(n, u))}
4      return QueryRoutingMechanism(w, q, K, l)
5  else
6      if Type(q) = "PinSearch" then
7          return GetObjectsFromIndexTable(K, −1)
8      else if one(u) ⊆ one(v) then  // i.e. SupersetSearch
9          objectsList ← GetObjectsFromIndexTable(K, l)
10         l ← l− Length(objectsList)
11         From(q) ← "Node"
12         while l > 0 do
13             c ← GetNextSBTreeChild(u)
14             cList ← QueryRoutingMechanism(c, q, K, l)
15             objectsList ← objectsList + cList
16             l ← l− Length(cList)
17         end
18         return objectsList
19     end
20 end
```

Figura 1.11: Algoritmo di routing [26]

1.5 Tecnologie correlate

1.5.1 Proof-of-Location

Una questione importante relativa all'utilizzo dei dati crowd-sensed riguarda la veridicità del dato stesso. I dati, infatti, volontariamente o involontariamente sono, spesso, incerti, imprecisi e inattendibili. Inoltre, vi è sempre il rischio che entità esterne possano manomettere e compromettere il dato. Dal

momento che, una volta memorizzati all'interno delle DLT, i dati non possono essere modificati, il momento critico si verifica, spesso, tra la fase di generazione del dato e quella di inserimento.
Esistono, tuttavia, diverse soluzioni che consentono di far fronte al problema; alcune di queste si basano su certificati che vengono rilasciati da entità attendibili e che, se allegati ai dati, consentono di verificare alcune proprietà [27].
La Proof-of-Location (PoL), per esempio, può essere definita come un certificato digitale che consente di dimostrare che l'utente si trovava in una certa posizione, in un determinato momento. La veridicità del certificato è garantita dalla fiducia riposta nell'autorità e dal mezzo di comunicazione (es. Wi-Fi o Bluetooth), in quanto limitato nel raggio di azione (garantendo quindi che due attori siano necessariamente vicini).
In generale, i certificati di PoL possono essere ottenuti attraverso i seguenti modi:

- **Autorità di certificazione e PKI**: per *Public key Infrastructure* (PKI), si intende un'infrastruttura che prevede l'utilizzo di certificati digitali, emessi da terze parti fidate che offrono il servizio di *Certification Authority*.

  Queste terze parti possono essere rappresentate da RSU o veicoli di trasporto pubblico che stabiliscono delle connessioni (tramite Wi-Fi o Bluetooth) con gli utenti che richiedono un certificato e, dopo opportuni controlli, ne rilasciano uno che attesta la presenza dell'utente in quella determinata posizione geografica.

- **Sistemi di localizzazione decentralizzati**: un esempio è FOAM [28],

un protocollo pensato per la PoL che sfrutta una rete permissionless e autonoma di radio beacons in grado di offrire servizi di localizzazione, indipendenti da sistemi centralizzati come il GPS. Al contrario del caso precedente, qui la fiducia viene riposta nella corretta esecuzione del protocollo decentralizzato quindi nel meccanismo di consenso.

I partecipanti alla rete sono incentivati a cooperare tra di loro, al fine di fornire la PoL agli utenti che la necessitano, usando metodi di geotriangolazione.

Tale sistema si basa, inoltre, su un nuovo standard di posizione, le coordinate cripto-spaziali (CSC), ossia informazioni di posizione basate su indirizzi Ethereum e geohash, accessibili dalle applicazioni decentralizzate.

- **Zero-Knowledge Proof of Location**: in crittografia, la *Zero-Knowledge Proof* è un metodo con cui una parte, detta *Prover*, può provare ad un'altra parte, detta *Verifier*, di conoscere un valore x senza dare alcuna informazione eccetto il fatto stesso di conoscere x.

  L'obiettivo è, dunque, quello di provare il possesso di un'informazione senza rivelare l'informazione stessa o qualsiasi informazione aggiuntiva.

  Similmente, il protocollo di *Zero-Knowledge Proof of Location* [29] consente ad un *verifier* di accertare che la posizione fornita dal *prover* si trovi all'interno o all'esterno di un'area senza che la posizione del prover stesso venga rivelata.

  A differenza degli altri due approcci, in questo caso è più facile garantire un livello di privacy più elevato.

1.5.2 Proof-of-Identity

Un altro tipo di verifica, spesso necessaria, è quella dell'identità degli utenti che utilizzano i servizi.

Di solito, i sistemi di gestione delle identità si basano su database centralizzati e gestiti dall'autorità che fornisce il servizio.

Esistono, tuttavia, diverse alternative decentralizzate basate sul modello di *Self-Sovereign Identity (SSI)* [30], in cui gli utenti dispongono del pieno controllo, ossia hanno la sovranità sulla loro identità e sulle informazioni da condividere.

Il concetto chiave delle SSI è rappresentato dalle "credenziali verificabili" che associano all'identità (DID) una serie di attributi e informazioni (claim). Dunque, in una sorta di rapporto trilaterale, si avrebbe l'utente che rappresenta il possessore delle credenziali conservate nel proprio wallet, il soggetto che emette le credenziali (l'ente pubblico o privato che certifica un determinato status) e, infine, colui al quale le credenziali sono presentate (es. un sistema informatico). Sotto questo nuovo paradigma sarà sempre il proprietario (titolare) della credenziale digitale ad avviare interazioni con servizi di terze parti, accettare o condividere dati [31]. Nel concretizzare tutto questo, le DLT giocano un ruolo centrale permettendo, a chi possiede un wallet apposito, di non delegare la custodia e il controllo delle informazioni personali a terze parti e decidere di esporre i certificati che più ritiene utili.

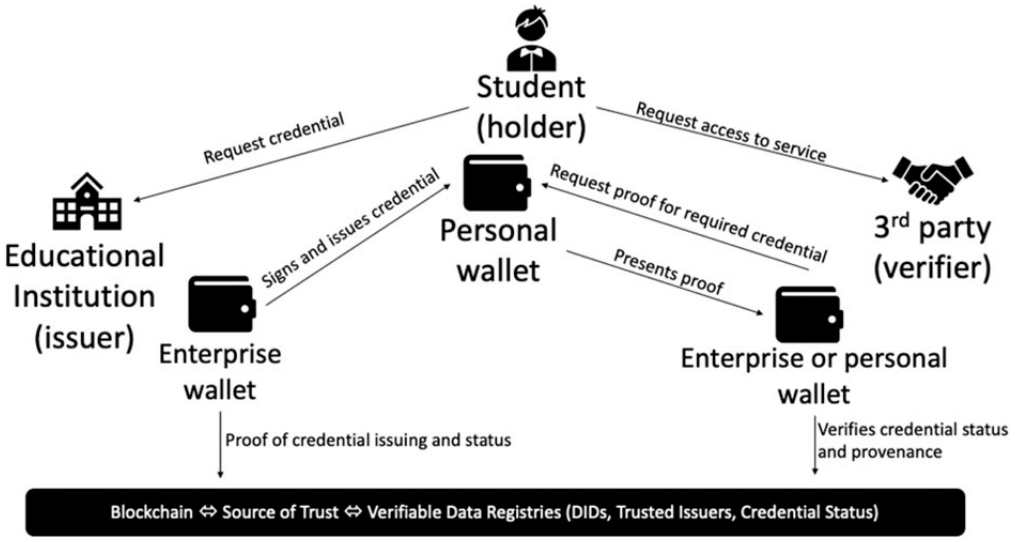

Figura 1.12: Componenti e flussi principali di uno scenario SSI [31]

1.5.2.1 EBSI

La *European Blockchain Services Infrastructure (EBSI)* [32] è un'infrastruttura DLT che si pone l'obiettivo di realizzare servizi pubblici a livello europeo, basati sulle tecnologie blockchain e caratterizzati da elevati livelli di sicurezza e privacy.

Tra le aree di applicazione affidate al progetto EBSI rientra anche quello dell'identità digitale, il cui scopo è quello di implementare un servizio di identità autonoma che consenta agli utenti di creare e controllare la propria identità in contesti transfrontalieri.

Questo caso d'uso è basato sull'*European Self-Sovereign Identity Framework* (ESSIF), un framework SSI esteso e adattato ai quadri normativi europei tra cui le direttive GDPR [31].

Un progetto simile è l'*Italian Blockchain Service Infrastructure (IBSI)*, un'iniziativa promossa da AgID che mira a realizzare la prima rete italiana basata

sulla blockchain per l'erogazione di servizi di interesse pubblico [33].

1.6 Lavori correlati

Le funzionalità e le potenzialità delle DLT sono sempre più riconosciute e apprezzate sia da aziende pubbliche che private. La loro capacità di consentire la verificabilità pubblica delle transazioni e dei dati digitali rappresenta, forse, la più importante, anche se non l'unica, ragione che le rende particolarmente adatte e sfruttabili in svariati contesti, quali IoT [34], [35], Smart Cities [36], [37] e ITS [26].

Relativamente all'ambito delle Smart Cities e della Smart Mobility, sono state proposte diverse architetture. Yuan e Wang nel loro lavoro [38] definiscono le basi per un nuovo modello di blockchain orientato agli ITS attenzionando, in particolar modo, la potenzialità delle blockchain di contribuire a stabilire un ecosistema sicuro, fidato e decentralizzato.

Zichichi, Ferretti e D'Angelo propongono un framework [39] completamente basato su tecnologie permissionless, pensato per creare, archiviare e condividere i dati generati dagli utenti mentre sono in movimento attraverso i sensori sui loro dispositivi e veicoli.

López e Farooq [40] presentano, invece, un modello basato su Hyperledger Fabric e focalizzato sulle problematiche connesse a privacy, sicurezza, gestione e scalabilità dei dati che sorgono durante la condivisione delle informazioni relative alla mobilità.

Diverse sono le proposte esistenti che condividono l'obiettivo del presente di lavoro, ossia fornire un contributo per incrementare la sicurezza sulle strade.

Overko e altri [41] propongono un sistema distribuito, basato su IOTA che sfrutta algoritmi di Reinforcement Learning per determinare una distribuzione sconosciuta dei modelli di traffico in una città.

Sempre sull'uso di IOTA negli ITS è da annoverare il lavoro di Bartolomeu e altri [42] che propongono l'uso di questa DLT per migliorare la sicurezza delle funzioni a bordo e all'esterno del veicolo. Altri lavori correlati si focalizzano sulla ricerca decentralizzata di dati all'interno delle DLT. A tal proposito, sono presenti diverse soluzioni di livello uno [43] e due [44].

Infine, è da segnalare l'uso sempre più massivo di sistemi di crowdsourcing basati su tecnologie Blockchain.

CrowdJury [45], per esempio, è un'applicazione di crowdsourcing, basata su blockchain per l'elaborazione delle sentenze in tribunale.

CrowdBC [46] è, invece, un framework decentralizzato in cui il compito di un richiedente può essere risolto da un gruppo di lavoratori senza fare affidamento su terze istituzioni di fiducia.

Capitolo 2

Architettura del sistema

L'obiettivo del presente lavoro è quello di costruire un sistema decentralizzato di raccolta e condivisione collaborativa delle informazioni relative ad ostacoli ed insidie presenti su strada, in cui i cittadini siano incentivati a fornire informazioni affidabili relative all'ambiente circostante, senza la presenza o l'intervento di un'autorità centrale. La progettazione di un tale sistema pone, inevitabilmente, una serie di requisiti che, in gran parte, le DLT permettono di soddisfare:

- **Privacy**: la sicurezza e la protezione della privacy sono state identificate come requisiti principali degli ITS. L'identità di chi è coinvolto nella transazione, infatti, deve essere necessariamente protetta. La natura crittografica delle DLT, che fa uso di un sistema a chiave pubblica/privata, fa sì che le transazioni siano pseudo-anonime [47]; dal punto di vista della privacy, dunque, in uno scenario di Smart Transportation, l'uso di una DLT è auspicabile. Inoltre, il controllo, diretto o indiretto, che gli individui esercitano attualmente sui loro dati personali è condizionato dalle

tecniche di gestione delle informazioni personali basate su piattaforme centralizzate. Queste sono spesso concentrate in pochi Internet Service Provider (ISP) che possono minare la privacy dei loro utenti [48].
L'uso delle DLT per gestire la condivisione e l'accesso ai dati in un sistema conforme alla normative come il GDPR [49] , può consentire la protezione dei dati personali degli utenti [50]. Infatti, il principale vantaggio di tali sistemi è che forniscono agli individui la possibilità di registrare i loro dati in alcuni spazi interoperabili di dati personali (PDS) [51], garantendo la sovranità dei dati senza l'intervento di fornitori centralizzati.

- **Persistenza e Tamper-proof**: una volta inviati e registrati nel sistema, i dati non devono poter essere alterati o rimossi da una singola entità. L'integrità e la disponibilità dei dati è garantita dalle DLT attraverso la loro replica tra tutti i nodi della rete e l'uso di tecniche crittografiche.

- **Microtransazioni**: fattori come l'enorme quantità di veicoli circolanti e la necessità di disporre di informazioni in tempo reale determinano l'esigenza di avere flussi di dati grandi e veloci.

- **Resilienza agli attacchi**: il sistema deve essere resistente agli attacchi e all'uso improprio da parte di attori malevoli. Esempi tipici sono gli attacchi di Spamming o la scrittura di informazioni false nel registro. Tutti questi casi possono essere notevolmente limitati da un uso combinato di un sistema di consenso basato su Proof-of-Work e Proof-of-Location.

- **Decentralizzazione**: gli utenti non devono dipendere da intermediari centralizzati che potrebbero influenzare il comportamento del sistema

in modo illecito o decidere regole che vadano a loro esclusivo vantaggio. L'uso delle DLT elimina la necessità che sia un'autorità centrale a elaborare, convalidare e autenticare le transazioni eseguite.

- **Assenza di barriere all'ingresso**: rispetto ad un database centralizzato, in mano ad un solo service provider, una DLT pubblica consente a chiunque di accedere ai dati. Questo favorisce la crescita di "piccole" aziende e servizi che non sono limitati da una barriera all'ingresso (es. pagare il provider per accedere ai dati).

2.1 ITS-oriented model

Una prima panoramica generale dell'architettura può essere descritta facendo riferimento al modello concettuale orientato agli ITS (Intelligent Transportation System), basato su tecnologia blockchain [38].
Tale framework è composto da 7 livelli, nello specifico:

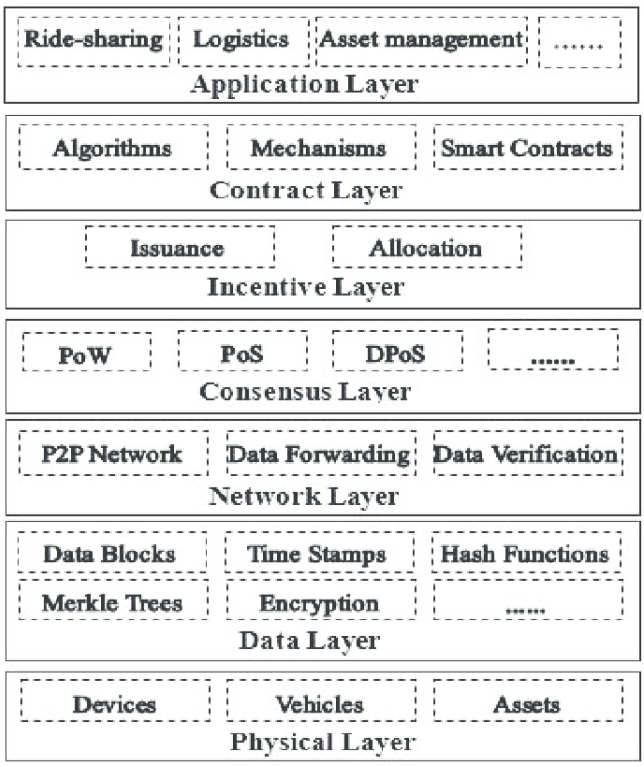

Figura 2.1: Modello concettuale a 7 strati orientato all'ITS [38]

- **Physical Layer**: questo primo livello incapsula vari tipi di entità fisiche quali dispositivi personali, veicoli e infrastrutture stradali, dotati di sensori e attuatori e in grado di stabile connessioni dirette e indirette tra di loro, nonché di raccogliere e trasmettere informazioni relative a parametri del veicolo, livelli di salute della persona e variabili ambientali.
  In questo specifico contesto, oggetto di interesse sono le informazioni di tipo geografico, ottenibili mediante tecnologia GPS.

- **Data Layer**: include il registro distribuito decentralizzato e le relative tecniche crittografiche tra cui algoritmi hash e Merkle Tree, fondamentali per rendere e mantenere i blocchi di dati sicuri. Come accennato precedentemente, il sistema in questione non si basa sul tradizionale mo-

dello di blockchain, bensì sul Tangle di IOTA, un particolar tipo di DLT specificatamente progettato per l'ambiente IoT.

- **Network Layer**: incorpora i meccanismi di rete distribuita, inoltro e verifica dei dati. I partecipanti alla rete, i peer, sono ugualmente privilegiati ed equipotenti senza alcun coordinatore centrale o struttura gerarchica; possono essere classificati in nodi *full* che possiedono, cioè, una copia completa dei dati e nodi *lightweight* che, al contrario, ne possiedono solo una piccola parte, ma possono richiedere i dati necessari ai nodi vicini, usando specifici protocolli.

- **Consensus Layer**: comprende i vari algoritmi necessari per raggiungere il consenso tra i nodi della rete circa la validità dei dati.
  Come si vedrà in seguito, il presente sistema fa uso di tre tipologie di consenso, Proof-of-Work, Proof-of-Location e Proof-of-Identity.

- **Incentive Layer**: incorpora i meccanismi di rewarding per i partecipanti alla rete, ovvero ricompense economiche che motivano i partecipanti a continuare gli sforzi al fine di mantenere la rete sicura, integra e affidabile.
  Come detto precedentemente, tale sistema si basa su IOTA, il cui modello organizzativo è basato più su una logica collaborativa che competitiva. L'assenza di fee per le transazioni e il fatto che in IOTA ogni nodo è un miner elimina la necessità di ricompensare i nodi per la creazione dei blocchi. Tuttavia, esistono altre forme di incentivi.
  Nel sistema in questione il meccanismo di rewarding è in funzione della quantità, della qualità e della tempistica delle informazioni condivise.

- **Contract Layer**: definisce il modo in cui gli utenti interagiscono tra di loro e con il sistema.

  Tale layer risulta composto da un insieme di smart contract, ossia protocolli informatici che facilitano, verificano e fanno rispettare la negoziazione o l'esecuzione dei contratti.

- **Application Layer**: specifica i potenziali scenari di applicazione e casi d'uso. Include i servizi decentralizzati messi a disposizione degli utenti, sfruttando funzionalità e dati prodotti nei livelli sottostanti.

2.2 Struttura del sistema

Il sistema in questione, adattamento del modello elaborato da M. Zichichi, L. Serena, S. Ferretti e G. D'Angelo [52], fa uso di un'aggregazione di diverse tecnologie distribuite che si incastrano perfettamente tra i blocchi del modello sopra descritto e che consentono la memorizzazione e la condivisione di dati crowd-sensed, cioè raccolti dagli utenti.
Nello specifico, il sistema adotta una soluzione di secondo livello che utilizza una DHT allo scopo di facilitare la ricerca di grandi quantità di dati.
Le soluzioni di "livello 2" si basano su reti o tecnologie che operano sopra le blockchain (livello 1) al fine di risolvere le difficoltà relative alla scalabilità.
Lo spostamento di una parte del carico transazionale dalla blockchain ad un'architettura ausiliaria permette di decongestionare il sistema e renderlo, in definitiva, più efficiente.
A livello architettonico si presenta come uno stack di layer:

- **DFS/DLT Network**: nodi che permettono ai client di collegarsi alla rete, leggere e scrivere informazioni
- **Files**: file o, in generale, dati mantenuti in memoria dai nodi
- **Keywords**: parole-chiave che descrivono i file e che ne consentono il recupero
- **Hypercube DHT**: sistema decentralizzato in cui vengono archiviati i file
- **DAO/Governance**: tecnologie e processi che costituiscono la governance della rete

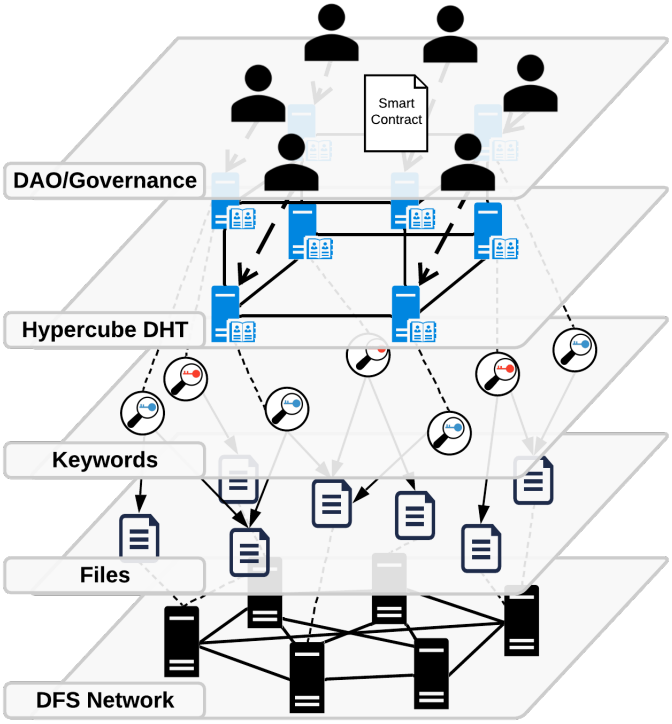

Figura 2.2: Architettura del sistema [52]

2.2.1 DFS/DLT Network

Questo primo livello comprende le tecnologie che si occupano dell'archiviazione dei dati e ne garantiscono disponibilità e affidabilità attraverso la replica. Due, in particolare, le tecnologie che possono essere utilizzate: DFS e DLT. Nella presente architettura il fattore discriminante per la scelta della tecnologia da utilizzare è la dimensione dei dati.

L'archiviazione in un Decentralized File System (DFS) richiede, in genere, latenze inferiori rispetto a quelle ottenibili tramite una DLT. Il solo utilizzo di un DFS, tuttavia, non è sufficiente per la convalida dei dati; pertanto, in caso di file di grandi dimensioni, sarebbe auspicabile integrare entrambe le tecnologie, per esempio, memorizzando i dati in un DFS (off-chain storage) e referenziandoli nella DLT (on-chain storage)[53].

Nel sistema proposto, comunque, l'utilizzo di una DLT, nello specifico IOTA, appare sufficiente, dal momento che i dati in questione sono semplici coordinate geografiche. Le DLT, oltre ad assicurare integrità, immutabilità e autenticità dei dati, permettono di evitare gli inconvenienti tipici degli approcci centralizzati, basati su server quali censura e single point of failure, rivelandosi una scelta ideale per la gestione e la condivisione di dati in contesti come quello in esame.

Tra le molteplici implementazioni di DLT la scelta di utilizzare IOTA è riconducibile in primo luogo alle aspettative in termini di prestazioni, nonché alla presenza del protocollo MAM.

Quest'ultimo permette la creazione e l'iscrizione a canali contenenti flussi di dati che confluiscono nel Tangle; tale protocollo rappresenta una valida alternativa alla semplice transazione IOTA, soprattutto, qualora si volesse rendere

i dati privati e, dunque, accessibili solo alle entità autorizzate.

2.2.2 Files

I dati acquisiti dai sensori, in questo specifico caso le coordinate dei punti in cui sono state rilevate insidie stradali, devono essere conservate in modo tale da poter essere recuperate al momento opportuno.
Ai nodi della rete IOTA spetta, dunque, il compito di archiviare e duplicare i dati in modo da decentralizzarne l'accesso. Gli ID delle transazioni o le root, nel caso si utilizzi MAM, sono il risultato del processo di archiviazione dei dati sul Tangle e rappresentano l'unico mezzo per identificare e recuperare l'oggetto; in altre parole, la chiave per l'accesso alle informazioni.
Questo meccanismo di riservatezza risulta particolarmente adatto in contesti in cui i dati sono creati al solo scopo di rimanere circoscritti al creatore o a un gruppo di utenti specifici, ma limitante, come in questo caso, in cui le informazioni sono destinate ad essere di pubblico dominio.

2.2.2.1 TAG

Una prima alternativa a cui si potrebbe pensare per far fronte al problema consiste nell'etichettare le transazioni, mediante il campo TAG, con le coordinate del luogo in cui è stata rilevata l'insidia. L'utilizzo di tale parametro consentirebbe, effettivamente, di effettuare ricerche sul Tangle dal momento che sarebbe possibile filtrare le transazioni che presentano un determinato TAG.
Questo meccanismo, tuttavia, non risolverebbe completamente il problema.

In primo luogo, l'utilizzo del TAG si rivela utile solo in caso di ricerca di informazioni specifiche. In questo contesto, invece, è necessario un meccanismo che consenta all'utente il recupero di informazioni relative non ad un singolo punto geografico, ma ad un'intera area (es. tutte le buche presenti nel quartiere San Donato di Bologna).
In secondo luogo, l'utilizzo di un TAG faciliterebbe l'alterazione di informazioni specifiche da parte di utenti malintenzionati ai quali basterebbe creare delle transazioni fittizie, inserendo un TAG corrispondente al set di dati che si vuole alterare.
I risultati che si otterrebbero dalla ricerca apparirebbero distorti e non più affidabili, compromettendo, così, la fiducia degli utenti nel sistema.
È necessario, dunque, una soluzione che permetta di effettuare ricerche che siano specifiche e generiche allo stesso tempo e che consenta di avere una sicurezza circa la veridicità dei dati che si ottengono.
La Distributed Hash Table permette di aggirare le problematiche appena descritte, consentendo specificità e generalità nelle ricerche.

2.2.3 Hypercube DHT

La DHT utilizzata è, essenzialmente, una struttura overlay che presenta una topologia a ipercubo, ottimizzata per l'esecuzione di query, basate su parole-chiave.
Come già accennato, i vantaggi di tale struttura risiedono in due fattori. In primo luogo ogni singolo oggetto (l'hash della transazione o la root del canale MAM) è indicizzato da un solo nodo, univocamente determinabile grazie alle keyword associate.

Inoltre, la particolare topologia a ipercubo fa sì che gli oggetti descritti da set di keyword simili siano gestiti da nodi vicini tra loro. Tale proprietà, oltre a velocizzare il routing tra nodi, risolve la questione della specificità delle informazioni, consentendo agli utenti del sistema di ottenere dati dettagliati relativi ad aree geografiche più o meno vaste.

2.2.4 Keywords

Nel sistema in questione, i contenuti vengono recuperati attraverso l'esecuzione di query sulla DHT basate sulla ricerca di parole-chiave associate ai dati.

Nello specifico, i dati sono rappresentati nella forma di coppie chiave-valore in cui il valore è costituito dall'oggetto inserito nella DHT, in questo caso l'hash della transazione IOTA, mentre la chiave è rappresentata dalle keyword associate all'oggetto in questione.

Nel momento in cui un oggetto viene inserito nella DHT occorre, quindi, specificare le keyword che meglio lo descrivono. Una corretta associazione delle keyword agli oggetti è, dunque, fondamentale per facilitare la ricerca e il recupero.

A titolo di esempio, se si volessero registrare i dati di temperatura e umidità dell'aria del quartiere di San Donato a Bologna, archiviati su un indirizzo MAM, tale oggetto potrebbe essere descritto da parole-chiave quali "temperatura", "umidità","San Donato", "Bologna", "Italia".

2.2.4.1 Codifica basata su OLC

Come già anticipato, i dati in questione sono di tipo geografico; nello specifico, si tratta di coordinate rappresentanti i punti in cui vengono rilevate le insidie stradali. Dunque, è necessario un meccanismo che consenta di associare in maniera automatica un set di keyword idoneo a descrivere in modo preciso una determinata posizione geografica.

Il meccanismo di generazione delle keyword, pensato per tale sistema, si basa su una doppia codifica.

La prima consiste nel semplificare la forma del dato di posizione, convertendolo semplicemente in *Open Location Code (OLC)* [54]. Questi ultimi rappresentano indirizzi stradali, estremamente precisi, di lunghezza simile a quella dei numeri telefonici, ma possono essere abbreviati in sole quattro o sei cifre. Meno cifre sono presenti, più è grande l'area considerata, e viceversa. Ad esempio, un codice di 4 cifre come 6P23 identifica un'area di 110 km.

Code length	2	4	6	8	+	10	11
Block size	20°	1°	0.05° (3′)	0.0025° (9″)		0.000125° (0.45″)	
Approximately	2200 km	110 km	5.5 km	275 m		14 m	3.5 m

Figura 2.3: Corrispondenza tra il numero di cifre di un OLC e ampiezza dell'area geografica [54]

La seconda codifica è un particolare tipo di conversione logica che consente di trasformare, in maniera univoca, l'OLC relativo ad una certa posizione geografica in un set di keyword.

Per esempio, dato l'OLC "6P000000+", è possibile immaginare che ogni carattere rappresenti una determinata keyword. Associando le keyword "6" e "P" a

posizioni bitstring, “000001” e “000010” rispettivamente, allora la conversione di “6P000000+” che le contiene entrambe, sarà “000011”.

Inoltre, l’associazione tra le keyword e le posizioni deve essere univoca; ciò significa che le keyword associate agli OLC rappresentanti aree più vaste debbano essere presenti anche negli OLC rappresentati aree più piccole.

Per esempio, le keyword associate ad un OLC relativo ad un’area di 2200 km (es. 6P000000+) dovranno essere presenti anche nel set di keyword associate all’OLC relativo ad un’area di 3,5 metri (es. 6PH57VP3+PR).

Quindi, supponendo che “000011” sia la codifica di “6P000000+”, una possibile codifica di “6PH57VP3+PR”, che contiene il prefisso “6P”, sarà “101011”.

La lunghezza della bitstring dipenderà dal livello di precisione che si sceglie di utilizzare per la codifica in OLC.

Per esempio, utilizzando OLC di 6 caratteri, si possono rappresentare aree di circa 5.5 km. Dal momento che gli OLC hanno una rappresentazione in base 20, per poter rappresentare in maniera univoca ogni possibile combinazione di un OLC di 6 caratteri, sono necessari 20^6 valori in binario, ovvero stringhe binarie di 25 caratteri.

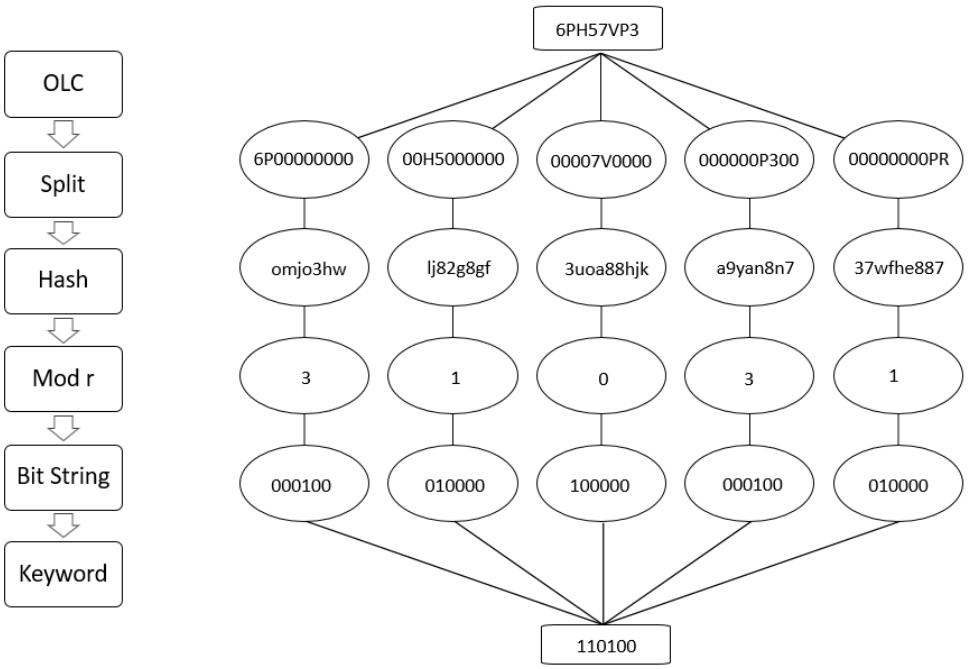

Figura 2.4: Processo di codifica con r=6

2.2.5 DAO Governance

Nel modello preso a riferimento, l'ultimo livello è rappresentato da un'organizzazione autonoma decentralizzata (DAO), costituita dai nodi della rete DHT e principalmente basata sull'uso degli smart contract che permettono alla stessa community di gestire un progetto, definendone i vari aspetti e partecipando attivamente alla governance.

In una DAO tutte le decisioni e le azioni intraprese vengono affrontate dall'intera community attraverso un sistema di votazione, senza la presenza di un'autorità centrale che ne controlli l'esito.

Oltre agli smart contract e al protocollo di consenso, il terzo meccanismo che permette il corretto funzionamento di una DAO è rappresentato dall'uso di to-

ken che consentono agli utenti di acquisire potere di voto e, allo stesso tempo, fungono da meccanismo di scambio e ricompensa economica.

Oltre alle DAO, altri strumenti indipendenti che possono risiedere nell'ultimo livello sono le applicazioni decentralizzate (DApps). Queste ultime, che si basano principalmente su Ethereum, sono anch'esse costituite da una serie di smart contract e pensate per un caso d'uso specifico.

Si distinguono dalle normali applicazioni per il fatto di essere completamente autonome e non soggette a controllo e blocco da parte del governo o della società. In altre parole, le DApps permettono di stabilire una connessione diretta tra utenti e servizi, dove gli utenti hanno il pieno controllo sulle informazioni e sui dati che condividono.

Capitolo 3

Use Case

3.1 Fasi

L'esecuzione del sistema si struttura in tre macro-operazioni:

- Segnalazione
- Notifica
- Rewarding

3.1.1 Segnalazione

La fase di segnalazione è rappresentata in figura 3.1 e si articola in 4 step.

Acquisizione del dato

Il primo, fondamentale, passo consiste nella creazione del dato; nel lavoro di J. Gandhi e altri [55] viene fornita una panoramica completa delle tecniche e

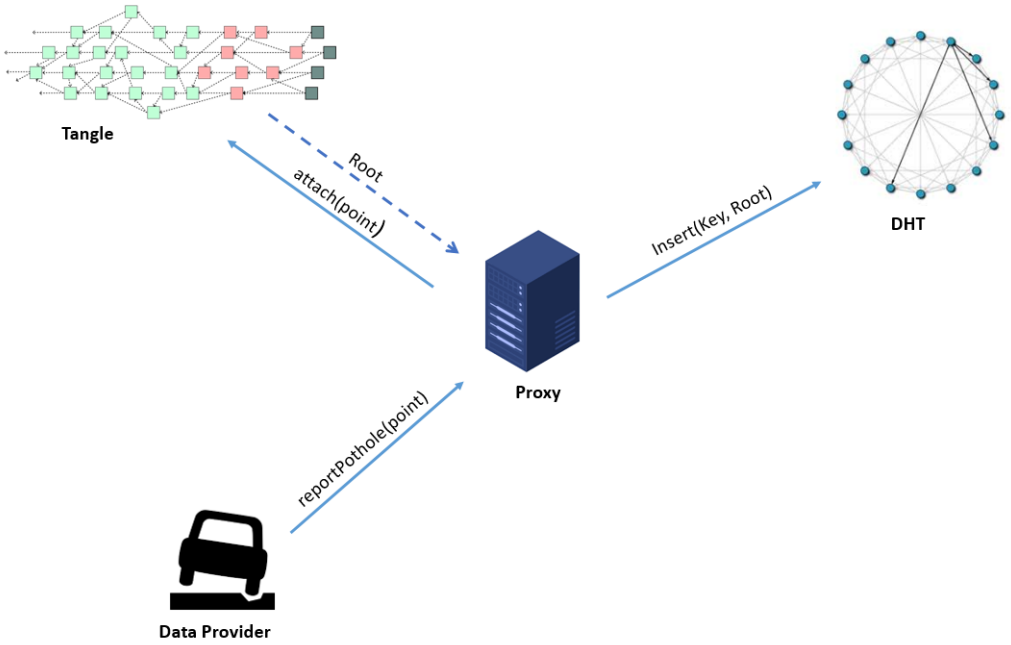

Figura 3.1: Fase di segnalazione

delle tecnologie utilizzate per l'identificazione delle anomalie stradali.
A livello generale, è possibile distinguere tali tecniche in due macro categorie:

- **Manuali**: non necessitano di hardware e software particolarmente costosi in quanto è l'utente stesso che agisce da sensore, segnalando la presenza di insidie tramite fotografie o indicandone la posizione geografica. I dati raccolti vengono poi elaborati utilizzando, in genere, varie tecniche di Machine Learning.

- **Basate su sensori**: utilizzano giroscopio, accelerometro e GPS degli smartphones per rilevare e misurare le vibrazioni che si manifestano in presenza di buche e dossi.

Nel presente lavoro la parte relativa all'identificazione delle anomalie non è stata implementata; tuttavia, si ipotizza la presenza di un'applicazione basata

sulla seconda tecnica, in cui i dati in questione vengono generati tramite i sensori presenti nelle AU (Application Unit) o nei veicoli [53].
In quest'ultimo caso, i dati vengono trasferiti alle AU attraverso sistemi di comunicazione intraveicolari (es. Bluetooth, USB, Wi-Fi).

Verifica del dato

La qualità e l'affidabilità del dato sono garantite attraverso un uso combinato di diverse tecniche. La Proof-of-Location, ottenibile tramite una delle tecniche già descritte in precedenza, aiuterebbe a prevenire i falsi rapporti di localizzazione delle posizioni (facilmente falsificabili con app GPS false), verificando la correttezza della dichiarazione dell'utente di trovarsi in una determinata posizione in un certo momento.
Inoltre, la Proof-of-Identity permetterebbe l'identificazione degli utenti che utilizzano il sistema in maniera decentralizzata. In questo caso, il sistema richiederebbe all'utente, al momento dell'iscrizione al servizio, la "prova" relativa a un qualche dato (es. informazioni contenute nella carta di identità) presente nel suo wallet, tramite l'uso delle credenziali verificabili. Il sistema SSI verificherebbe la veridicità della dichiarazione, trasmettendo al richiedente non i dati, bensì la prova della presenza di tali dati.
Infine, la Proof-of-Work, che i partecipanti della rete devono svolgere in fase di segnalazione per inserire il dato sul Tangle, assicurerebbe protezione contro eventuali attacchi di spam.
La combinazione di queste "prove" consente la creazione di un servizio decentralizzato non necessitante di un'entità centrale che supervisiona le interazioni.

Inserimento nel Tangle

Il dato raccolto viene inserito nel Tangle tramite semplici transazioni IOTA oppure utilizzando il protocollo MAM.
Per questo specifico caso d'uso i dati oggetto di interesse sono solo quelli di tipo geografico. Tuttavia, qualora i dati raccolti fossero di vario genere (per esempio temperatura, stato del veicolo ecc), l'uso del protocollo MAM si rivelerebbe più appropriato in quanto ciascuna tipologia di dato verrebbe pubblicata su un canale diverso. Tutti i canali (feature channel) risulterebbero poi indicizzati da un index channel, reso pubblico a chiunque debba accedere alle informazioni [53].

Inserimento nella DHT

L'ID della transazione (o la root nel caso si utilizzi MAM), acquisito in seguito all'inserimento dei dati nel Tangle, viene indicizzato nella rete DHT associando un set di keyword che ne consente il successivo recupero. Quest'ultimo è determinato dal processo di codifica descritto nel presente lavoro in 3.2.
La DHT, infatti, non mantiene i dati del messaggio, ma solo l'indirizzo che ne permette l'individuazione all'interno del Tangle. Questo fa sì che la proprietà di integrità dei dati, propria dei sistemi DLT, non venga violata.

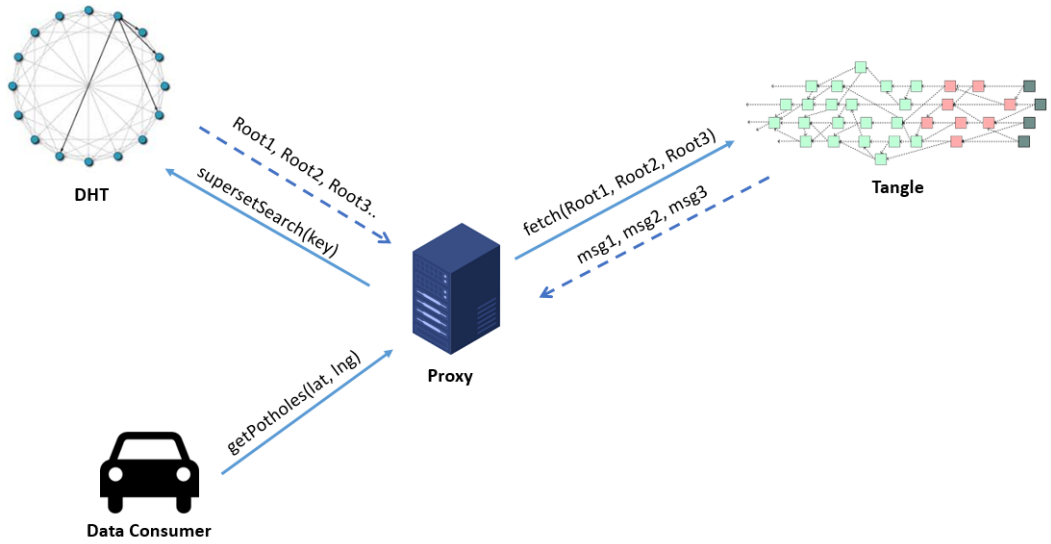

Figura 3.2: Ricerca dei dati

3.1.2 Notifica

Ricerca

La seconda fase del sistema è rappresentare in figura 3.2.

Data una certa posizione, si vogliono ottenere tutte le segnalazioni di anomalie stradali presenti all'interno di quella precisa area.

Contrariamente a quanto avviene nella fase di inserimento, in questo caso il sistema dialoga prima con la DHT. La differenza tra le due tipologie di ricerca che è possibile fare sulla DHT, Pin e Superset, risiede, fondamentalmente, nel numero di oggetti che si possono ottenere.

- **Pin Search**: in questo caso viene contattato esclusivamente il nodo che gestisce il keyword set in input, ottenuto mediante lo stesso processo di codifica utilizzato nella fase di inserimento. L'output di tale ricerca è, dunque, costituito da tutti e solo quegli oggetti associati a quel particolare keyword set.

- **Superset Search**: questa tipologia di ricerca può essere considerata un'estensione della Pin Search in quanto gli oggetti restituiti provengono sì dal nodo che gestisce il set di keyword specificato, ma anche da nodi che rappresentano altri keyword set. In questo caso, quindi, faranno parte del risultato tutti quegli oggetti che possono essere descritti, seppur non strettamente, dal keyword set indicato dall'utente.

  Ciò significa ottenere tutte le anomalie stradali presenti non solo in prossimità di una specifica posizione, ma all'interno di un'area più o meno vasta.

  L'ampiezza dei risultati ottenibili fa sì che le richieste alla DHT possano essere fatte meno frequentemente, il che riduce la possibilità che un eccessivo numero di richieste possa sovraccaricare il sistema.

  Com'è facile intuire, l'estensione dell'ambito della Superset Search la rende più adatta a questo specifico caso d'uso rispetto alla ristrettezza di campo della Pin Search.

Utilizzo del dato

Una volta recuperati, i dati, in formato geospaziale, vengono visualizzati in tempo reale sulle mappe presenti sui dispositivi degli utenti, fornendo supporto alla navigazione.

Grazie alla ricerca Superset, infatti, l'utente verrà avvisato della presenza di un'anomalia sulla strada molto prima di trovarsi nel punto esatto in cui è stata effettuata la segnalazione, avendo, dunque, tempo a sufficienza per rallentare nelle immediate vicinanze, evitarla o prendere una percorso alternativo.

Oltre agli automobilisti, ad avere accesso ai dati sono gli addetti alla manu-

tenzione delle strade che operano per conto dell'amministrazione cittadina o di enti privati che, oltre a verificare la veridicità della segnalazione, avranno la possibilità di intervenire tempestivamente prevenendo incidenti.

3.1.3 Rewarding

Gli utenti che svolgono il lavoro di raccolta e mappatura dei dati geospaziali vengono ricompensati attraverso token in maniera simile a quanto descritto in [56].

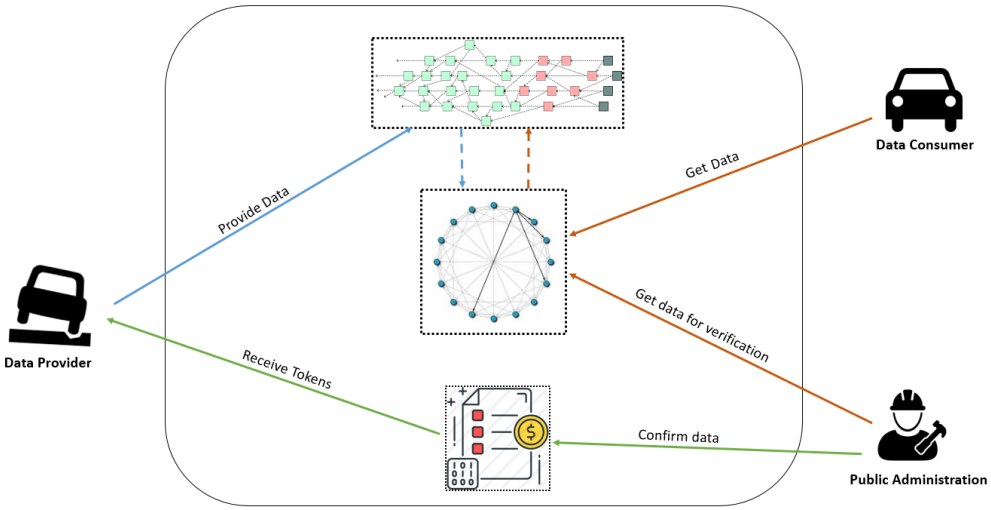

Figura 3.3: Rappresentazione grafica del sistema

Una volta confermata l'effettiva presenza dell'insidia e verificate la PoL e la PoI, i token vengono trasferiti nei wallet degli utenti che possono, poi, utilizzarli per il pagamento di altri servizi gestiti o convenzionati con l'ente pubblico (es. pedaggi autostradali, parcheggi ecc).

Il meccanismo di ricompensa potrà basarsi su una qualche strategia compensativa proporzionata al merito (es. i primi ad aver effettuato la segnalazione

vengono premiati con una quantità di token maggiore).

Il trasferimento dei token avviene tramite esecuzione degli smart contract che tracciano tutte le condizioni in base alle quali è possibile riconoscere compensi o autorizzare l'erogazione di servizi.

Questo sistema di ricompensa incentiverebbe la partecipazione degli utenti, favorendo l'impegno e la cooperazione [57] [58].

Capitolo 4

Performance Evaluation

Questa parte è dedicata alla valutazione della bontà del sistema. Nello specifico, dato il caso d'uso per il quale è pensato, è presumibile ipotizzare che il sistema venga utilizzato da numerosi utenti; pertanto, la fattibilità dipende, in gran parte, da quanto i meccanismi e le tecnologie impiegate nelle fasi di inserimento e ricerca dei dati si rivelino adatti, in termini di efficacia ed efficienza, alla complessità presente in contesti reali.

4.1 Componenti principali

L'obiettivo dei test effettuati è, dunque, quello di fornire un quadro, quanto più preciso, relativo alle prestazioni delle due principali tecnologie utilizzate, IOTA e la DHT, il tutto, attraverso l'utilizzo di scenari che si avvicinino, il più possibile, alla realtà.
Nel caso della DHT, il software è implementato in Python ed espone le quattro azioni principali dei nodi utilizzando il framework server Flask, cioè inserimen-

to di un oggetto, rimozione di un oggetto, ricerca Pin e ricerca Superset.
I nodi logici, ciascuno dei quali eseguiti da un server Flask dedicato, dopo una fase di bootstrap, si collegano ai loro vicini in base alla topologia a ipercubo.
Per quanto riguarda IOTA, per la connessione e la comunicazione con il Tangle, sono state utilizzate le librerie Javascript.
Relativamente al protocollo MAM, per semplicità si è scelto di utilizzare un canale di tipo public, con seed casuale [59] [60].

4.2 Strumenti

Tutti i test in questione sono stati eseguiti utilizzando un PC Intel Core i7-1065G7 CPU, 16 GB RAM.
Per quanto riguarda IOTA sono state testate entrambe le modalità per l'inserimento dei dati sul Tangle, ovvero tramite semplici transazioni e utilizzando il protocollo MAM. Inoltre, sono stati impiegati diversi nodi che differiscono per il tipo di rete in cui si trovano.
Nello specifico, la Mainnet è il network ufficiale in cui i token hanno valore monetario. Relativamente alla Mainnet, i provider utilizzati sono i seguenti:

- **nodo privato**: 1 core CPU, 2 GB RAM, 50GB di storage, VPS `https://www.digitalocean.com/products/droplets/`
- **nodi pubblici**: `https://chrysalis-nodes.iota.org`

Al contrario, la Testnet è una rete adibita al testing in cui i token, essendo simulati, non hanno valenza reale. In quest'ultima la PoW è meno dispendiosa,

infatti, il suo livello di difficoltà, rappresentato dal Minimum Weight Magnitude è impostato a 9, molto meno rispetto al 14 richiesto per la Mainnet.
Relativamente alla Testnet sono stati utilizzati i seguenti provider:

- **IOTA single transaction**: `https://api.lb-0.testnet.chrysalis2.com`
- **MAM**: `https://nodes.devnet.iota.org` [1]

I risultati dei test sono mostrati tramite grafici creati utilizzando la libreria Python Matplotlib. Nello specifico, i grafici che mostrano i tempi di latenza sono generati utilizzando campioni casuali dei dati prodotti.

4.3 Scenari

4.3.1 Elementi

Due primi elementi fondamentali per la creazione di scenari realistici sono la presenza di veicoli e le strade che questi percorrono. Inoltre, considerando il caso d'uso del sistema, ossia la rilevazione di insidie stradali, un terzo elemento necessario è la presenza di buche/ostacoli lungo i percorsi. È, altresì, necessario considerare un aspetto di complessità, ossia la presenza di traffico che permetta, in un certo senso, di stressare il sistema valutando, conseguentemente, la sua capacità di adattamento a situazioni reali.
Per quanto riguarda la simulazione degli scenari, sono state seguite le seguenti fasi:

[1] Al momento dei test è stato utilizzato il protocollo 1.0, non più supportato dalla Mainnet. Pertanto, ai fini del confronto con MAM è stata utilizzata esclusivamente la Testnet.

- Creazione di 10 percorsi, rappresentati attraverso liste di coordinate geografiche, ciascuno dei quali avente almeno un punto (latitudine, longitudine) in comune con un altro;
- Definizione della classe *Vehicle* in modo tale da poter istanziare dei tipi di auto, a seconda del percorso seguito;
- Codifica e memorizzazione nella DHT dei punti in comune tra le liste di coordinate, rappresentanti le insidie stradali;
- Simulazione del movimento su strada, una volta istanziati i veicoli, scorrendo gli elementi della lista di coordinate del relativo percorso. Il valore del delay time, utilizzato per scorrere la lista, determina la velocità con la quale il sistema viene interrogato;

4.3.2 Test setup

Per ciascun percorso, sono stati istanziati 10 veicoli e avviati simultaneamente. Al fine di stressare la DHT, le simulazioni sono state effettuate utilizzando un delay time di soli 3 minuti tra l'invio di una query e la successiva anche se, in un contesto reale, l'utilizzo della ricerca Superset consentirebbe di ottenere dati aggregati relative ad aree vaste, pertanto, la frequenza delle interrogazioni sarebbe relativamente bassa.
Per entrambe le tipologie di ricerca, Pin e Superset, sono state testate diverse configurazioni della rete; nello specifico è stato utilizzato un parametro r pari a 3, 4, 5, 6 corrispondente, cioè, ad una rete composta da 8, 16, 32 e 64 nodi. Una volta determinato il keyword set, cioè effettuando la codifica da coppia di coordinate a OLC prima e da OLC a bitstring dopo, la query è indirizzata

ad un nodo della DHT scelto casualmente. Da qui, attraverso il meccanismo di routing, viene raggiunto il nodo responsabile del set di keyword specificato che, in caso di ricerca Pin, restituisce tutti gli oggetti associati ad esso. Nel caso della ricerca Superset, invece, è stata impostata una soglia massima l pari a 10 di oggetti che vengono restituiti all'utente.

4.4 Fase di segnalazione

4.4.1 Latenza di IOTA

Un primo elemento da considerare nella fase di segnalazione è la latenza relativa all'inserimento dei dati nel Tangle che dipende, principalmente, da due sottoprocessi:

- **Selezione delle tip**: approvazione di due precedenti transazioni
- **Proof-of-Work**: esecuzione della PoW per validare la transazione

La PoW può essere effettuata localmente sul dispositivo, prima di inviare la transazione al nodo, o in remoto, ossia dallo stesso nodo IOTA.
Se da una parte, eseguita in remoto permette effettivamente di risparmiare potenza computazionale, dall'altra, però, c'è sempre il rischio che, a seconda della potenza e del numero di richieste che riceve, il nodo IOTA potrebbe non avere sufficienti risorse per completarla in un lasso di tempo soddisfacente o, nel peggiore dei casi, non completarla affatto.
Per la selezione delle tip, invece, è necessario possedere una visione completa del Tangle, pertanto, questo processo è sempre eseguito da un Full Node.

Un primo confronto, dunque, riguarda i periodi di latenza ottenuti dai nodi della Mainnet e della Testnet svolgendo la PoW localmente. I risultati sono mostrati in figura 4.1.

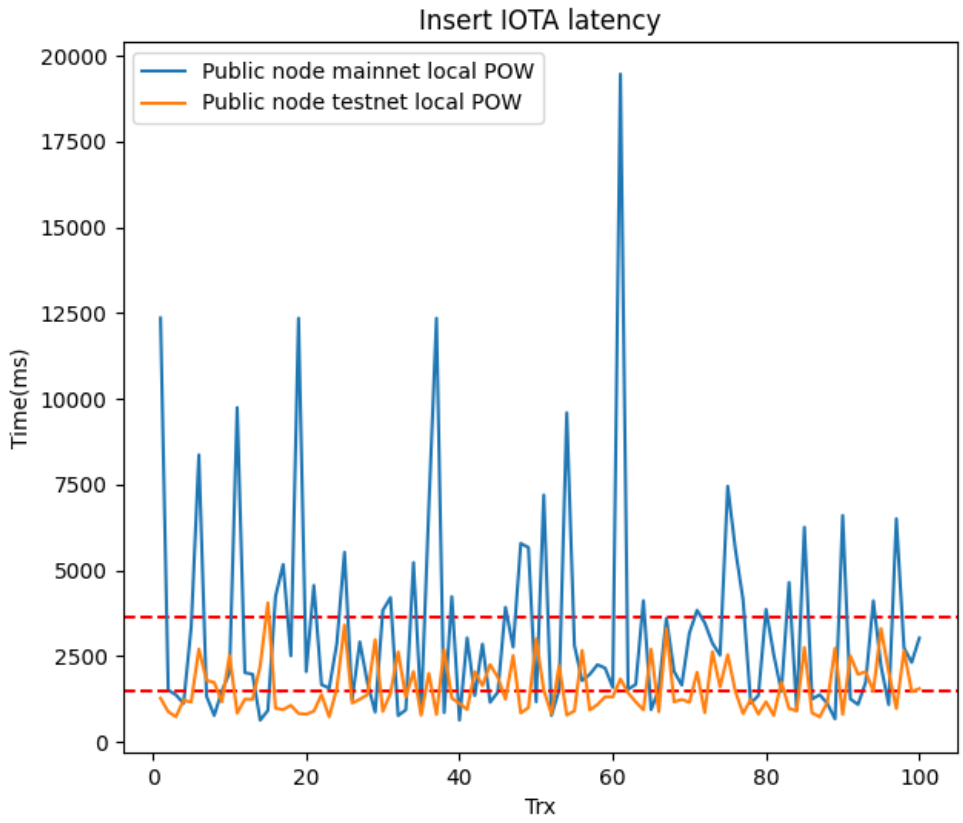

Figura 4.1: Latenza nel processo di inserimento dei dati nel Tangle, utilizzando la Testnet e la Mainnet

È da notare come i tempi di latenza dei nodi della Mainnet siano molto più variabili rispetto a quelli dei nodi della Testnet. Il che è dovuto, probabilmente, al tempo necessario per la selezione delle tip e alla maggiore difficoltà della PoW. Tuttavia, nel caso della Mainnet, la tempistica richiesta per l'inserimento di una transazione risulta, in media, relativamente bassa (3.63 secondi),

sebbene sia un pò più del doppio di quella richiesta nel caso della Testnet (1.48 secondi).

Un altro aspetto rilevante riguarda la differenza in termini di modalità di svolgimento della PoW. In figura 4.2 sono mostrati i tempi di inserimento ottenuti effettuando la PoW in locale e in remoto.

In primo luogo, dai grafici si evince come, in entrambi i casi, sebbene l'inserimento delle transazioni con PoW da remoto appaia un pò più veloce, la differenza tra le due modalità non è poi così accentuata.

Si tenga presente, però, che per ottenere tali risultati sono necessari nodi con hardware adeguati; in assenza di questi, delegare l'onere della PoW al nodo IOTA è l'opzione migliore.

Per quanto riguarda, invece, il confronto tra il nodo privato e i nodi pubblici, prendendo in considerazione i valori di latenza medi più bassi, ovvero quelli ottenuti optando per la PoW in remoto, si nota come la differenza tra i due sia quasi trascurabile; infatti, se da una parte i nodi pubblici, ricevendo più transazioni, sono in grado di fornire le tips più velocemente, dall'altra, però, potrebbero avere meno risorse computazionali da fornire ai client per lo svolgimento della PoW. Questo si traduce, chiaramente, in un aumento della latenza nel processo di inserimento della transazione.

Sebbene i due nodi presentino un tempo medio di inserimento simile, nel caso del noto privato, la varianza è molto più grande. Questo è chiaramente dovuto alla quantità minore di risorse computazionale rispetto al nodo pubblico, infatti, con lo stesso carico di lavoro, i risultati del nodo privato sono molto più altalenanti.

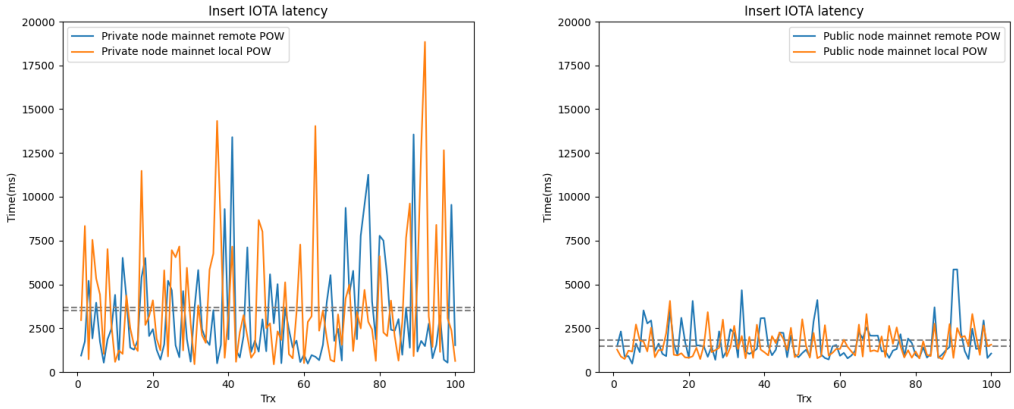

Figura 4.2: Confronto della latenza nel processo di inserimento dei dati nel Tangle, effettuando la PoW in locale e in remoto

Un ultimo elemento da attenzionare riguarda la modalità di pubblicazione del dato.

I risultati, visibili in figura 4.3, mostrano che l'utilizzo del protocollo MAM richiede un tempo medio di latenza sensibilmente inferiore rispetto a quello previsto utilizzando una semplice transazione IOTA. Come detto precedentemente, però, nell'implementazione è stato possibile usare solo la Testnet, in cui la difficoltà della PoW è notevolmente ridotta. L'utilizzo della Mainnet, infatti, avrebbe comportato una latenza sicuramente maggiore dovuta al fatto che, per trasmettere un messaggio MAM, sono necessarie tre intere transazioni, dunque, più tip e più PoW.

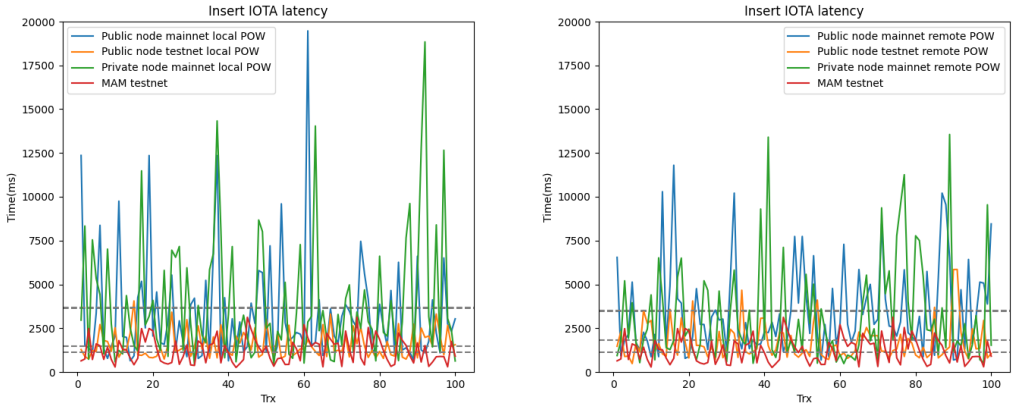

Figura 4.3: Confronto della latenza nel processo di inserimento dei dati nel Tangle, utilizzando il nodo pubblico, il nodo privato e il protocollo MAM

La tabella 4.1 riassume i risultati relativi alla latenza nel processo di inserimento delle transazioni sul Tangle. Nello specifico, vengono riportati i valori della media e dell'intervallo di confidenza al 95%, calcolati per ogni test e divisi per il numero di transazioni effettuate.

IOTA latency insert				
Node	**PoW**	**Net**	**Avg. Latency (sec)**	**Conf.Int.(95%)**
Private	Remote	Mainnet	3.52	[3.21,3.82]
Private	Local	Mainnet	3.69	[3.53,3.86]
Public	Remote	Mainnet	3.48	[3.22,3.74]
Public	Local	Mainnet	3.63	[3.32,3.94]
Public	Remote	Testnet	1.8	[1.68,1.92]
Public	Local	Testnet	1.48	[1.07,1.21]
MAM	Remote	Testnet	1.1	[1.01,1.18]

Tabella 4.1: Valori di latenza nel processo di inserimento dei dati nel Tangle

4.4.2 Latenza della DHT

Il tempo impiegato per l'inserimento dei dati viene attenzionato anche per quanto riguarda la DHT.

Una volta inserito il dato su IOTA, la query di inserimento viene indirizzata ad un nodo della DHT, scelto in maniera casuale. Se il keyword set K dell'oggetto, ossia il risultato del processo di codifica, coincide con l'ID del nodo contattato, quest'ultimo si occuperà della memorizzazione del dato, viceversa, in un processo ricorsivo, il messaggio verrà inoltrato al nodo vicino fino a quando verrà raggiunto l'effettivo nodo responsabile.

Il tempo di latenza dipende, pertanto, dall'efficienza del meccanismo di routing della DHT, nonché dalla dimensione della rete.

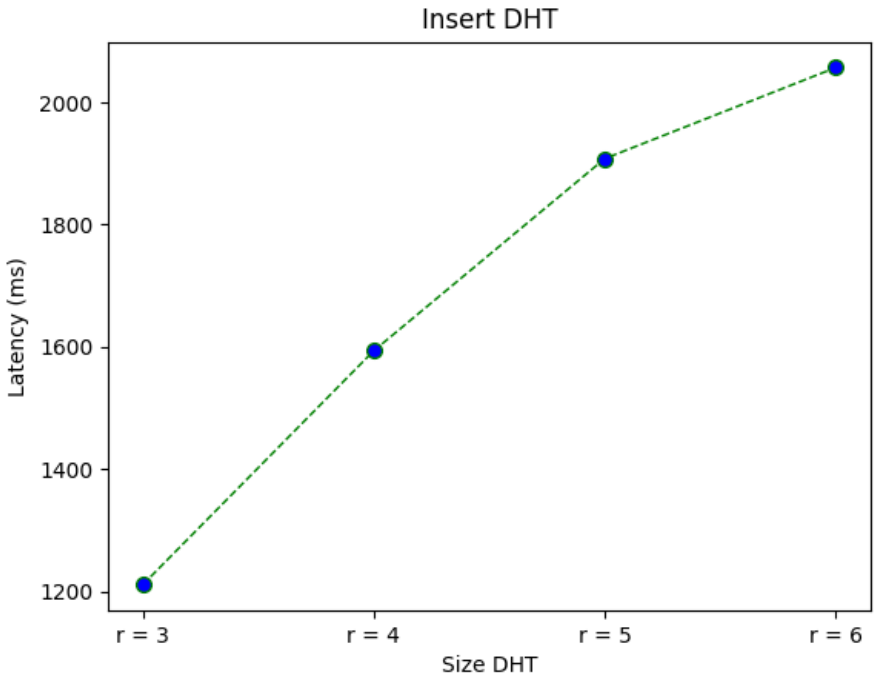

Figura 4.4: Latenza media nel processo di inserimento dei dati nella DHT

In figura 4.4 si nota, infatti, come il tempo medio di inserimento dell'oggetto nella DHT cresca in funzione del parametro r che determina la dimensione dell'ipercubo.

Variando da poco più di un secondo, con un r=3, a poco più di due con un r=6, il tempo medio di inserimento risulta complessivamente basso.

Un secondo aspetto, visibile in figura 4.5, riguarda la presenza di outlier in tutte le simulazioni effettuate; infatti, in alcuni casi l'inserimento nella DHT richiede tempi maggiori rispetto alla media.

La presenza di anomalie, la cui frequenza aumenta all'aumentare della dimensione della rete, è, in parte, riconducibile al fattore di casualità nella scelta del nodo che riceve la richiesta. Infatti, se l'ID del nodo iniziale risulta molto diverso dal set di keyword K, potrebbero volerci molti inoltri prima di raggiungere l'effettivo responsabile; questo si traduce, chiaramente, in un aumento del tempo di latenza.

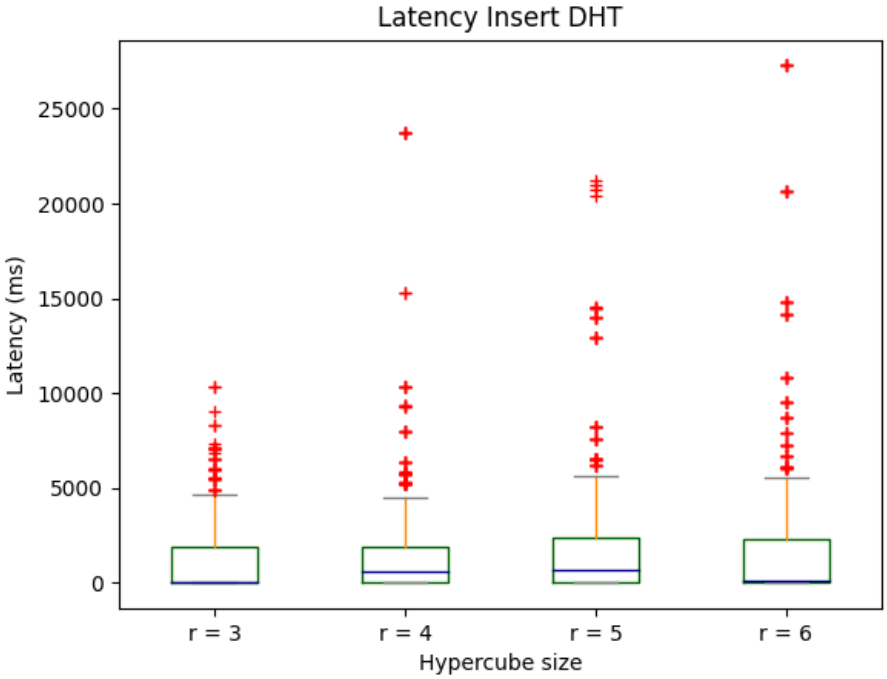

Figura 4.5: Presenza di outliers nel processo di inserimento nella DHT

4.5 Fase di ricerca

4.5.1 Numero di hop

Per quanto riguarda la fase di ricerca, un primo criterio di valutazione è quello dell'efficienza del meccanismo di routing della DHT che è alla base del processo di ricerca per parole-chiave.

Di seguito vengono mostrati i risultati relativi alla Pin Search e alla Superset Search.

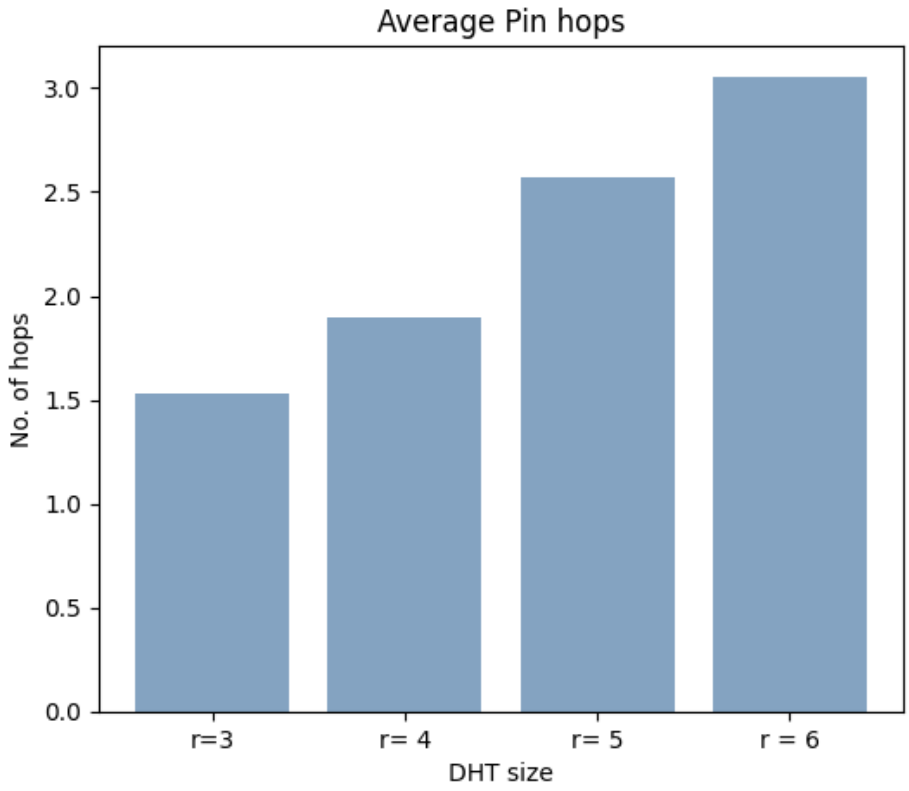

Figura 4.6: Media del numero di hop nel caso della Pin Search

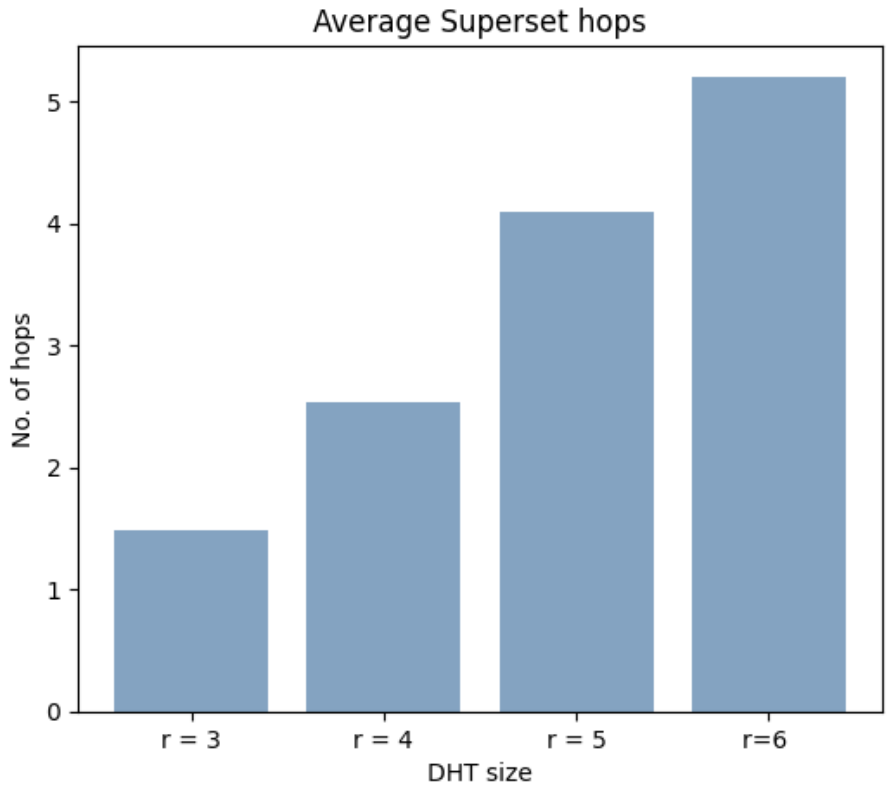

Figura 4.7: Media del numero di hop nel caso della Superset Search

Come ci si aspetterebbe, in entrambi i casi, il numero di hop necessari per trasmettere un messaggio dal nodo di partenza a quello di destinazione cresce all'aumentare della dimensione dell'ipercubo. Infatti, più aumenta il numero di nodi, più si allunga il percorso che un messaggio deve fare per arrivare a destinazione.

Pin Search

Relativamente alla ricerca Pin, il numero medio di hop è dell'ordine del logaritmo di numero di nodi logici, cioè $\frac{log(n)}{2}$ o $\frac{r}{2}$. Infatti, incrementa da 1.53 secondi, con r=3, a 3.04 secondi, con r=6.

Superset Search

Teoricamente, nel caso della ricerca Superset, il numero medio di hop dovrebbe essere $\frac{log(n)}{2} + l$, cioè, uguale alla media degli hop necessari per arrivare al nodo responsabile dell'insieme di parole-chiave della query K, più la media degli hop per arrivare da quel nodo a tutti i nodi che includono K, fino al raggiungimento del limite di oggetti. La crescita quasi lineare, visibile in figura 4.7, è dovuta al fatto che, in alcune ricerche, sono necessari più hop del previsto per raggiungere il nodo di destinazione.

La presenza di outlier, che influisce sulla media complessiva si verifica, perlopiù, nelle fasi iniziali delle simulazioni e in corrispondenza di valori maggiori del parametro r.

Questo fenomeno può essere riconducibile al fatto che la ricerca Superset attraversa il Binomial Spanning Tree del sub-ipercubo indotto dal nodo responsabile dell'insieme di parole-chiave, fino a trovare il numero di oggetti indicato dal limite, in questo caso $l = 10$. Pertanto, quando i nodi della rete sono molti e gli oggetti che questi archiviano, al contrario, sono pochi, molti nodi risultano vuoti, cioè non fanno riferimento ad alcun oggetto. In tale situazione è, dunque, necessario un maggior numero di salti per soddisfare le ricerche.

Una volta archiviato un numero sufficiente di oggetti, il numero di hop tende a stabilizzarsi, rientrando nella media.

4.5.2 Latenza della DHT

In figura 4.8 si nota che l'aumento della dimensione dell'ipercubo influisce in maniera minima sulla latenza nel recupero degli oggetti. Infatti, in entrambe le tipologie di ricerca, indipendentemente dal numero di nodi, il tempo medio di ricerca, in una situazione relativamente complessa come quella ipotizzata nella creazione degli scenari, risulta complessivamente basso.
È da notare, inoltre, come le due tipologie di ricerca differiscano poco in termini di latenza media, sebbene nel caso della Superset Search, sia necessario un tempo maggiore dovuto all'esplorazione dello Spanninng Binomial Tree.

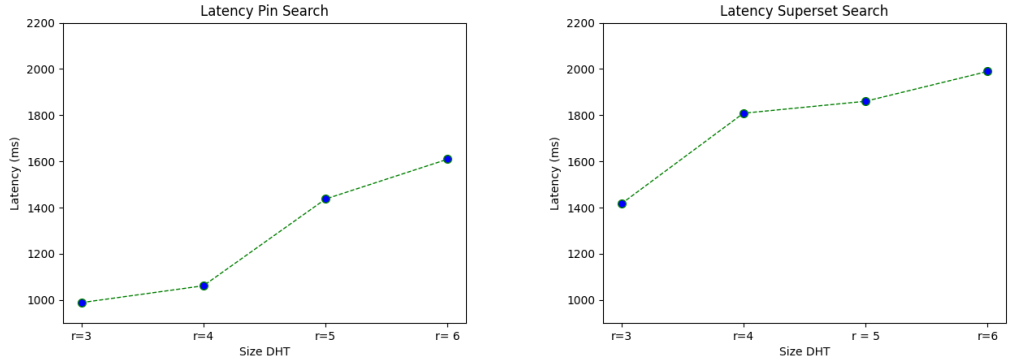

Figura 4.8: Latenza media nei processi di ricerca Pin e Superset

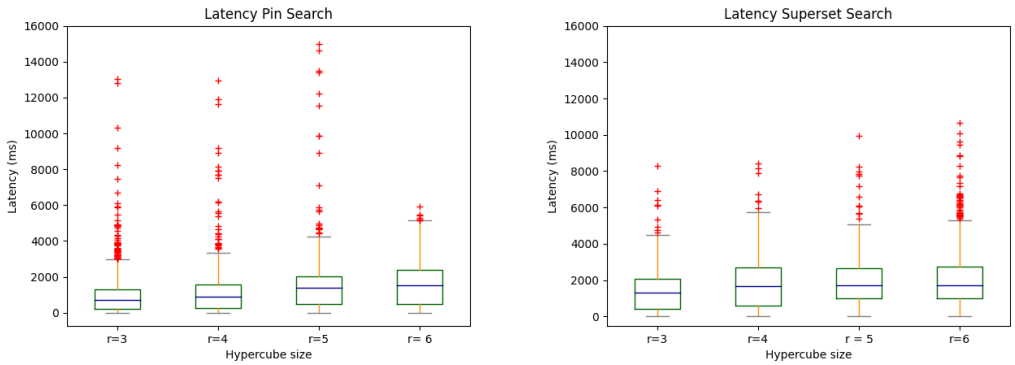

Figura 4.9: Presenza di outliers nei processi di ricerca Pin e Superset

4.5.3 Latenza di IOTA

Come fatto nella fase di inserimento del dato, ossia valutare la latenza di IOTA, allo stesso modo, anche in questa fase di ricerca occorre tenere conto del tempo impiegato nel recupero dei dati sul Tangle.
Un primo test riguarda il confronto tra i nodi delle due reti. A differenza di quanto accade nell'inserimento, nel caso del recupero, utilizzando i nodi pubblici della Testnet, si ottiene un tempo medio di latenza significativamente maggiore rispetto a quello ottenuto nel caso della Mainnet.

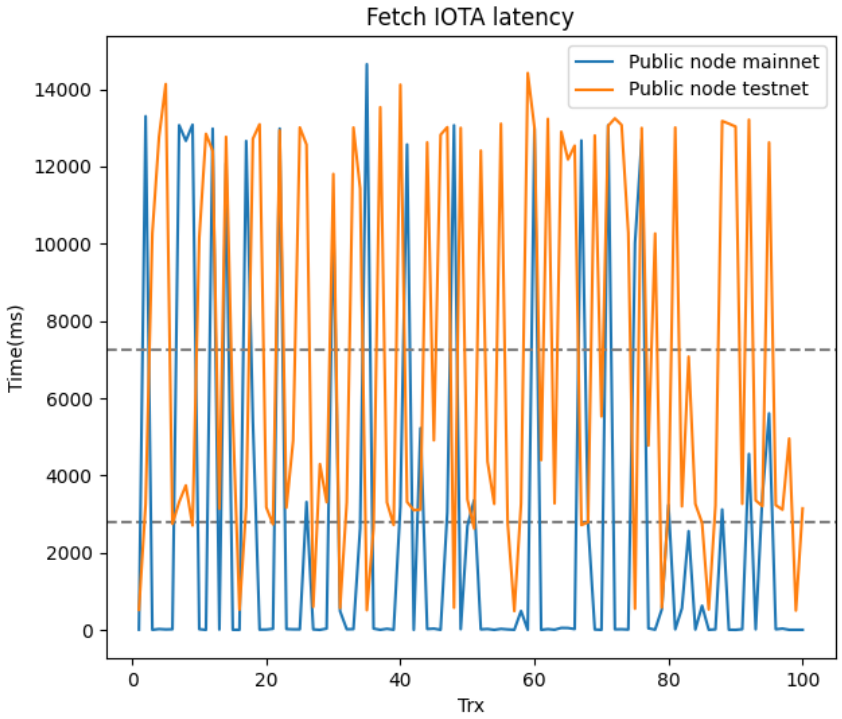

Figura 4.10: Confronto della latenza relativa al recupero dei dati, utilizzando i nodi pubblici della Testnet e della Mainnet

Confrontando, invece, i risultati dei nodi pubblici con quelli ottenuti utiliz-

zando il nodo privato, si nota una differenza trascurabile in termini di latenza media, tuttavia, nel primo caso le variazioni temporali sono molto più instabili.

Infine, l'utilizzo della modalità MAM si rivela, ancora una volta, l'opzione migliore dal momento che la latenza media si riduce ulteriormente.

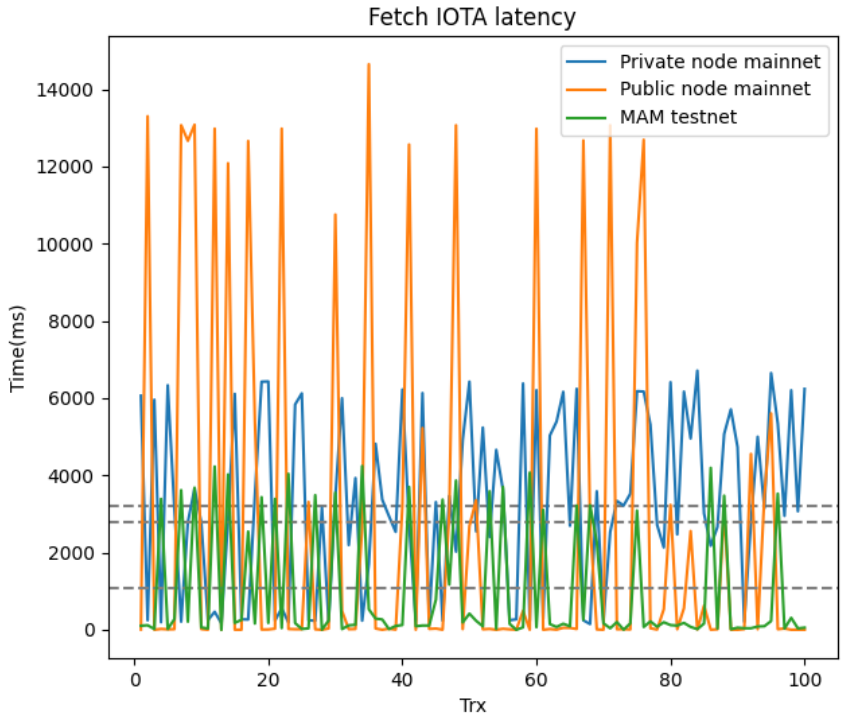

Figura 4.11: Confronto della latenza relativa al recupero dei dati utilizzando il nodo pubblico, privato della Mainnet e il protocollo MAM

Di seguito si riassumono i risultati della latenza relativi alla fase di recupero dei dati.

IOTA latency fetch			
Node	**Avg. Latency (sec)**	**Net**	**Conf.Int.(95%)**
Private	3.2	Mainnet	[3.15,3.26]
Public	2.81	Mainnet	[2.53,3.08]
Public	7.24	Testnet	[6.96,7.53]
MAM	1.1	Testnet	[1.01,1.18]

Capitolo 5

Discussione

Quest'ultima parte viene dedicata ad alcune considerazioni riguardanti il sistema presentato in questa tesi, con particolare riferimento alle tecnologie utilizzate.

5.1 IOTA

L'assenza di fee, il potenziale in termini di scalabilità e la struttura del registro sono solo alcune caratteristiche che rendono IOTA adatto al contesto dell'IoT e a nuovi paradigmi quali quello di Machine-to-Machine (M2M).
Nel presente sistema, così come in molte applicazioni di Smart Transportation, IOTA assume un ruolo centrale nel processo di scambio e memorizzazione delle informazioni. In entrambe le fasi, quella di inserimento e quella di recupero dei dati, i test eseguiti mostrano, in termini di latenza, risultati soddisfacenti, sebbene migliorabili.
Relativamente alla fase di segnalazione, i tempi di latenza non sembrano subire

particolari cambiamenti a seconda della modalità di svolgimento della PoW. Questo dipende, però, dalle risorse del dispositivo che si utilizza.

Lo stesso vale per la tipologia di nodo impiegato. I risultati dei test mostrano, infatti, che la differenza in termini di prestazioni tra i nodi pubblici e il nodo privato è quasi trascurabile anche se, in uno scenario reale, andrebbe considerato il trade-off tra velocità e risorse computazionali. Se è vero, infatti, che l'utilizzo dei nodi pubblici, soprattutto se molto usati all'interno della rete, permetta di accelerare il processo di selezione delle tip, dall'altra, questi, per via delle troppe richieste, potrebbero risultare carenti in termini di risorse computazionali. Questo può rappresentare un problema nei casi in cui si renda necessario delegare al nodo IOTA l'onere della PoW. Infatti, è da considerare la possibilità che i dispositivi degli utenti e le unità di calcolo presenti sui veicoli possano non avere sufficienti risorse computazionali.

Sebbene su molti aspetti risulti vincente, IOTA presenta ancora dei limiti quali la centralizzazione del cosiddetto Coordinatore (sezione 1.2.4), ovvero un singolo nodo speciale gestito dalla IOTA Foundation, ideato per proteggere il Tangle da possibili attacchi.

La presenza del Coordinatore appare in parziale contrasto con il concetto di decentralizzazione, dal momento che, in linea di principio, dà alla IOTA Foundation l'ultima parola sullo status quo della rete. Inoltre, un attacco al coordinatore potrebbe paralizzare l'intero Tangle, per non parlare del fatto che la sua esistenza limita la scalabilità di IOTA, che è uno dei più grandi punti di forza del protocollo.

Il Coordinatore è, tuttavia, un elemento temporaneo destinato ad essere rimosso attraverso il cosiddetto Coordicide, già in fase di lancio, dopo di che la

rete IOTA diverrà definitivamente decentralizzata.
Un altro progetto in fase di sviluppo è Qubic, la soluzione proposta da IOTA per la computazione quorum-based e l'implementazione di smart contract. Mediante l'utilizzo di questo protocollo potranno essere sviluppate vere e proprie applicazioni distribuite, come accade, al momento, per le dApp sviluppate in Ethereum.

5.2 DHT

Nelle DLT tradizionali, come Ethereum e IOTA, la ricerca di un dato all'interno dei registri è un processo lungo e complesso e la soluzione correntemente adottata è quella di utilizzare degli esploratori DLT centralizzati. Il presente sistema adotta, invece, una soluzione di livello 2 che fa uso di una DHT costruita sopra la DLT, in questo caso IOTA, con l'obiettivo di facilitare la ricerca di grandi quantità di dati.
La DHT utilizzata ha come peculiarità l'avere una struttura a ipercubo; questa particolare topologia permette, tramite l'esecuzione di query basate su parole-chiave, un accesso facile, veloce e decentralizzato ai dati memorizzati su IOTA. Ciascun nodo della rete è responsabile di uno specifico insieme di parole-chiave, derivato dal suo ID. Sia per l'inserimento che per la ricerca di informazioni nella DHT, i messaggi, etichettati opportunamente da un insieme di keyword, vengono scambiati di nodo in nodo, attraverso un particolare meccanismo di routing, fino a che il messaggio non arriva al destinatario, ossia il nodo che gestisce quel determinato insieme di keyword.
L'efficienza della DHT è stata valutata attraverso due parametri, il tempo di

latenza delle operazioni e la media dei salti necessari per localizzare i contenuti. Relativamente al primo caso, le tempistiche richieste per l'inserimento e per la ricerca dei dati risultano complessivamente basse. Inoltre, la differenza in termini di latenza tra le due tipologie di ricerca appare minima, sebbene la ricerca Superset richieda un tempo maggiore dovuto all'esplorazione dello Spanning Binomial Tree del nodo radice.
Per quanto riguarda il numero medio di salti, in entrambe le tipologie di ricerca, questo aumenta all'aumentare della dimensione della rete per poi stabilizzarsi in corrispondenza di un numero maggiore di oggetti archiviati. Ad ogni modo, il numero medio di scambi tra i nodi per far giungere, così come per recuperare, il messaggio risulta relativamente basso; infatti, rientra nell'ordine di O(log N), dove N indica il numero di nodi logici dell'ipercubo. Tali risultati confermano la capacità della DHT di essere utilizzata in scenari, oltre che reali, complessi.

5.3 Sviluppi futuri

MOBI è un'iniziativa nata nel Maggio del 2018 sotto forma di organizzazione no-profit per migliorare la mobilità attraverso l'uso della blockchain e delle tecnologie correlate ai registri distribuiti. MOBI conta sulla partecipazione di una serie di attori che operano su diverse componenti del mondo mobility (tra cui Ford, Daimler Benz, BMW, Renault, IBM, Accenture, VW, IOTA e Hyperledger) uniti dalla volontà di rendere la mobilità sempre più sicura, sostenibile e accessibile.
Essa agisce come una comunità "neutrale", nella quale le aziende sviluppano soluzioni innovative e standard in collaborazione con università, fondazioni pri-

vate ed altri tipi di organizzazioni tramite la creazione e condivisione di “proof of concept”.

A tal proposito un possibile sviluppo del progetto potrebbe implicare l’integrazione del sistema proposto con gli standard rilasciati da MOBI e la partecipazione nella fase di definizione e sviluppo di alcuni di questi tramite le esperienze consolidate dal presente lavoro.

Conclusioni

In questo lavoro è stato presentato un sistema di crowdsourcing decentralizzato e ideato per il contesto di Smart Transportation, che fa uso di un robusto schema di indicizzazione di dati, pensato come alternativa ai tipici schemi centralizzati, sempre più frequentemente oggetto di attacchi e uso improprio.
Interamente basato su tecnologie distribuite, oltre a fornire tracciabilità e immutabilità dei dati, il sistema in questione fa uso di una Distributed Hash Table come livello posto sopra la DLT, al fine di migliorare la gestione dei dati in un contesto in cui l'attenzione degli utenti è in progressivo incremento e dove, inoltre, efficacia e efficienza si rivelano più che importanti, necessari.
Le aspettative in termini di prestazioni, la struttura del registro e, soprattutto, l'assenza di commissioni per le transazioni effettuate sono solo alcuni aspetti che giustificano la scelta di utilizzare IOTA come DLT. D'altro canto, la DHT, con la sua particolare topologia a ipercubo, permette di semplificare notevolmente i processi di ricerca e acquisizione dei dati con ovvi vantaggi in termini di scalabilità, velocità e tolleranza ai guasti.
L'integrazione di questi due livelli unita ad altre tecnologie e paradigmi decentralizzati, quali gli smart contract e la Self Sovereign Identity, consente di dar vita a sistemi aperti, neutrali e sicuri che sfuggono al controllo di enti centrali,

considerato che il potere risiede nei nodi, e quindi negli utenti stessi.
Il caso d'uso ipotizzato è quello della rilevazione di insidie stradali, in cui gli utenti contribuiscono, attraverso la produzione di dati, a migliorare la sicurezza sulle strade. L'architettura del sistema si rivela comunque potenzialmente adattabile ad altre situazioni; pertanto, prendendo spunto da questa Proof-of-Concept, è possibile la realizzazione di altre applicazioni.
Infine, relativamente alla valutazione del sistema, dai test effettuati emerge come l'uso di IOTA richieda alcune considerazioni inerenti al contesto di utilizzo quali la tipologia di nodo da utilizzare e la modalità di svolgimento della PoW. Per quanto riguarda l'uso della DHT, in entrambe le tipologie di ricerca, il numero medio di salti e la latenza delle varie operazioni confermano la capacità di rispondere in maniera efficace, veloce e precisa alle interrogazioni anche in contesti potenzialmente complessi.

Bibliografia

[1] L. Figueiredo, I. Jesus, J. Machado, J. Ferreira, and J. Martins de Carvalho, "Towards the development of intelligent transportation systems," in *ITSC 2001. 2001 IEEE Intelligent Transportation Systems. Proceedings (Cat. No.01TH8585)*, 2001, pp. 1206–1211.

[2] M. Kawser, S. Sajjad, S. Fahad, S. Ahmed, and H. Rafi, "The perspective of vehicle-to-everything (v2x) communication towards 5g," *IEEE Access*, vol. 19, pp. 146–155, 04 2019.

[3] M. B. Mollah, J. Zhao, D. Niyato, Y. L. Guan, C. Yuen, S. Sun, K.-Y. Lam, and L. H. Koh, "Blockchain for the internet of vehicles towards intelligent transportation systems: A survey," *IEEE Internet of Things Journal*, vol. 8, no. 6, pp. 4157–4185, 2021.

[4] G. Vella, "Distributed ledger technology: Definizione e caratteristiche." [Online]. Available: https://blog.osservatori.net/it_it/distributed-ledger-technology-significato

[5] "Hyperledger fabric." [Online]. Available: https://www.hyperledger.org/use/fabric

[6] N. Szabo, "Formalizing and securing relationships on public networks," *First Monday*, vol. 2, no. 9, Sep. 1997. [Online]. Available: https://firstmonday.org/ojs/index.php/fm/article/view/548

[7] Ethereum. [Online]. Available: https://ethereum.org/en/

[8] Bitcoin. [Online]. Available: https://bitcoin.org/it/

[9] K. Werbach, "Trust, but verify: Why the blockchain needs the law," *Cyberspace Law eJournal*, 2017.

[10] Ethereum, "Erc-20 token standard." [Online]. Available: https://ethereum.org/en/developers/docs/standards/tokens/erc-20/

[11] ——, "Erc-721 non-fungible token standard." [Online]. Available: https://ethereum.org/en/developers/docs/standards/tokens/erc-721/

[12] IOTA. [Online]. Available: https://www.iota.org/

[13] S. Popov, O. Saa, and P. Finardi, "Equilibria in the tangle," *Computers and Industrial Engineering*, vol. 136, pp. 160–172, 2019. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0360835219304164

[14] A. Capossele, S. Müller, and A. Penzkofer, "Robustness and efficiency of voting consensus protocols within byzantine infrastructures," *Blockchain: Research and Applications*, vol. 2, no. 1, p. 100007, 2021. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S2096720921000026

[15] [Online]. Available: https://hackernoon.com/how-iota-solves-blockchains-scalability-problem-12e5cae05531

[16] S. Popov, "The tangle," 2018. [Online]. Available: https://assets.ctfassets.net/r1dr6vzfxhev/2t4uxvsIqk0EUau6g2sw0g/45eae33637ca92f85dd9f4a3a218e1ec/iota1_4_3.pdf

[17] [Online]. Available: https://whitepaper.io/document/3/iota-whitepaper

[18] C. Ye, G. Li, H. Cai, Y. Gu, and A. Fukuda, "Analysis of security in blockchain: Case study in 51

[19] I. Foundation, "The coordicite," 2020. [Online]. Available: https://files.iota.org/papers/20200120_Coordicide_WP.pdf

[20] V. Kaartemo and M. Kramer, *The sources of cybersecurity threats in cryptocurrency*. Routledge, 10 2020.

[21] M. Colavita and G. Tanzer, "A cryptanalysis of iota ' s curl hash function," 2018.

[22] I. F. Blog, "Introducing mam," 2021. [Online]. Available: https://blog.iota.org/introducing-masked-authenticated-messaging-e55c1822d50e/

[23] [Online]. Available: https://medium.com/coinmonks/iota-mam-eloquently-explained-d7505863b413

[24] Y.-J. Joung, C.-T. Fang, and L.-W. Yang, "Keyword search in dht-based peer-to-peer networks," in *25th IEEE International Conference on Distributed Computing Systems (ICDCS'05)*, 2005, pp. 339–348.

[25] F. Mazzini, "Query sull'infrastruttura iota: un approccio keyword-based," 2020. [Online]. Available: https://amslaurea.unibo.it/20603/

[26] M. Zichichi, L. Serena, S. Ferretti, and G. D'Angelo, "Towards decentralized complex queries over distributed ledgers: a data marketplace use-case," 04 2021.

[27] M. Zichichi, "A distributed ledger based infrastructure for intelligent transportation systems," 2018. [Online]. Available: https://amslaurea.unibo.it/18440/1/zichichi_mirko_tesi.pdf

[28] Foam, "Foam whitepaper," 2018. [Online]. Available: https://foam.space/publicAssets/FOAM_Whitepaper_May2018.pdf

[29] W. Wu, E. Liu, X. Gong, and R. Wang, "Blockchain based zero-knowledge proof of location in iot," in *ICC 2020 - 2020 IEEE International Conference on Communications (ICC)*, 2020, pp. 1–7.

[30] G. Kondova and J. Erbguth, "Self-sovereign identity on public blockchains and the gdpr," New York, NY, USA, p. 342–345, 2020. [Online]. Available: https://doi.org/10.1145/3341105.3374066

[31] A. Grech, I. Sood, and L. Ariño Martin, "Blockchain, self-sovereign identity and digital credentials: Promise versus praxis in education," *Frontiers in Blockchain*, vol. 4, 03 2021.

[32] C. Digital, "Ebsi." [Online]. Available: https://ec.europa.eu/cefdigital/wiki/display/CEFDIGITAL/EBSI

[33] "Blockchain: Agid promotrice dell'infrastruttura italiana ibsi." [Online]. Available: https://www.agid.gov.it/it/agenzia/stampa-e-comunicazione/notizie/2021/03/02/blockchain-agid-promotrice-dellinfrastruttura-italiana-ibsi

[34] X. Wang, X. Zha, W. Ni, R. Liu, Y. Guo, X. Niu, and K. Zheng, "Survey on blockchain for internet of things," *Computer Communications*, vol. 136, 01 2019.

[35] A. Reyna, C. Martín, J. Chen, E. Soler, and M. Díaz, "On blockchain and its integration with iot. challenges and opportunities," *Future Generation Computer Systems*, vol. 88, pp. 173–190, 05 2018.

[36] K. Biswas and V. Muthukkumarasamy, "Securing smart cities using blockchain technology," pp. 1392–1393, 2016.

[37] S. Ibba, A. Pinna, M. Seu, and F. Pani, "Citysense: blockchain-oriented smart cities," pp. 1–5, 05 2017.

[38] Y. Yuan and F.-Y. Wang, "Towards blockchain-based intelligent transportation systems," in *2016 IEEE 19th International Conference on Intelligent Transportation Systems (ITSC)*, 2016, pp. 2663–2668.

[39] M. Zichichi, S. Ferretti, and G. D'angelo, "A framework based on distributed ledger technologies for data management and services in intelligent transportation systems," *IEEE Access*, vol. 8, pp. 100 384–100 402, 2020.

[40] D. López and B. Farooq, "A multi-layered blockchain framework for smart mobility data-markets," *Transportation Research Part C: Emerging Technologies*, vol. 111, p. 588–615, Feb 2020. [Online]. Available: http://dx.doi.org/10.1016/j.trc.2020.01.002

[41] R. Overko, R. Ordóñez-Hurtado, S. Zhuk, P. Ferraro, A. Cullen, and R. Shorten, "Spatial positioning token (sptoken) for smart mobility," *IEEE Transactions on Intelligent Transportation Systems*, pp. 1–14, 2020.

[42] P. C. Bartolomeu, E. Vieira, and J. Ferreira, "Iota feasibility and perspectives for enabling vehicular applications," 2018.

[43] P. Jiang, F. Guo, K. Liang, J. Lai, and Q. Wen, "Searchain: Blockchain-based private keyword search in decentralized storage," *Future Gener. Comput. Syst.*, vol. 107, pp. 781–792, 2020.

[44] "The graph." [Online]. Available: https://thegraph.com/

[45] F. Ast and A. Sewrjugin, "Crowdjury, a crowdsourced justice system for the collaboration era," 11 2015.

[46] M. Li, J. Weng, A. Yang, W. Lu, Y. Zhang, L. Hou, J. Liu, Y. Xiang, and R. Deng, "Crowdbc: A blockchain-based decentralized framework for crowdsourcing," *IEEE Transactions on Parallel and Distributed Systems*, vol. 30, pp. 1251–1266, 2019.

[47] L. Tennant, "Improving the anonymity of the iota cryptocurrency," 2017.

[48] E. Pariser, *The filter bubble: What the Internet is hiding from you.* Penguin UK, 2011.

[49] Council of European Union, "Regulation (eu) 2016/679 - directive 95/46," pp. 1–88.

[50] M. Zichichi, S. Ferretti, and G. D'Angelo, "On the efficiency of decentralized file storage for personal information management systems," in *Proc. of the 2nd International Workshop on Social (Media) Sensing, co-located with 25th IEEE Symposium on Computers and Communications 2020 (ISCC2020).* IEEE, 2020, pp. 1–6.

[51] European Commission, "A european strategy for data," pp. 1–35, 2020.

[52] M. Zichichi, L. Serena, S. Ferretti, and G. D'Angelo, "Governing decentralized complex queries through a dao," 2021.

[53] M. Zichichi, S. Ferretti, and G. D'Angelo, "A distributed ledger based infrastructure for smart transportation system and social good," in *2020*

IEEE 17th Annual Consumer Communications Networking Conference (CCNC), 2020, pp. 1–6.

[54] "Open location code." [Online]. Available: https://en.wikipedia.org/wiki/Open_Location_Code

[55] J. Gandhi, U. Jaliya, and D. Thakore, "A review paper on pothole detection methods," *International Journal of Computer Sciences and Engineering*, vol. 7, pp. 379–383, 02 2019.

[56] A. Lisi, A. De Salve, P. Mori, L. Ricci, and S. Fabrizi, "Rewarding reviews with tokens: An ethereum-based approach," *Future Generation Computer Systems*, vol. 120, pp. 36–54, 2021. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0167739X21000480

[57] S. K. Bista, S. Nepal, and C. Paris, "Engagement and cooperation in social networks: Do benefits and rewards help?" in *2012 IEEE 11th International Conference on Trust, Security and Privacy in Computing and Communications*, 2012, pp. 1405–1410.

[58] R. Dolan, J. Conduit, J. Fahy, and S. Goodman, "Social media engagement behaviour: A uses and gratifications perspective," *Journal of Strategic Marketing*, vol. 24, 12 2015.

[59] daduz11, leuriniale96, "hypfs." [Online]. Available: https://github.com/daduz11/hypfs.git

[60] felap96, "IOTA_DHT." [Online]. Available: https://github.com/felap96/IOTA_DHT.git

Ringraziamenti

Ringrazio innanzitutto il Professor Ferretti e il Dottor Zichichi per gli indispensabili consigli, la gentilezza e la disponibilità dimostratemi lungo tutto il periodo di tirocinio e di tesi.

Ringrazio la mia famiglia, inclusa Ala, per avermi spronato ad andare avanti e a fare sempre di meglio. Se sono qui oggi lo devo soprattutto a loro.

Ringrazio i miei nonni, fonte inesauribile di affetto e di supporto morale.

Infine, ringrazio Manuel che mi è sempre stato accanto, aiutandomi nei momenti difficili e gioendo con me delle mie vittorie.