POLITECNICO DI TORINO

Corso di Laurea Magistrale

in Ingegneria Gestionale

Tesi di Laurea Magistrale

La tecnologia Blockchain nei processi di certificazione

Relatore:

Prof. Paolo Giaccone

Candidato:

Stefano Puttero

Aprile 2021

A tutti coloro che mi hanno sostenuto…

RINGRAZIAMENTI

Giunto al termine di questo percorso durato più di cinque anni vorrei spendere un paio di righe per ringraziare le persone che mi hanno sostenuto ed hanno reso possibile tutto questo.

In primis vorrei ringraziare il professore Paolo Giaccone, relatore della tesi di laurea. La ringrazio per la disponibilità ed attenzione che mi ha dedicato in questi mesi. Nonostante il particolare periodo storico in cui ci troviamo ho sempre sentito il suo supporto. Grazie per le lunghe consulenze che mi hanno guidato nella stesura dell'elaborato.

Un secondo ringraziamento va a BeChain, in particolare a Leonardo Mignone che mi ha supportato e fornito materiale utile per l'elaborato. In questo contesto un sentito grazie anche ad Iman che ha permesso l'analisi conclusiva dell'elaborato ed a Silvio Bilucaglia per la parte delle certificazioni.

Un grande ringraziamento lo voglio dedicare alla mia famiglia. Grazie al vostro continuo ed instancabile sostegno sono riuscito a concludere questo lungo percorso. Siete sempre stati al mio fianco.

Un ringraziamento speciale va a Lela. Sappiamo entrambi che gran parte di questo risultato è anche tuo. Mi hai sostenuto nei momenti più difficili e spronato al raggiungimento dell'obiettivo. Non dimenticherò mai tutto quello che hai fatto per me. Grazie davvero, sei la migliore compagna di viaggio che potessi desiderare.

Per ultimi, ma non per importanza, vorrei ringraziare tutti i miei amici e colleghi che ho incontrato in questi anni. In particolare, una dedica speciale ad un amico che purtroppo non c'è più. Sarai sempre al mio fianco a sostenermi.

Un grazie sentito a tutti.

INDICE

INTRODUZIONE

Alla base del seguente lavoro di tesi vi è lo studio della tecnologia blockchain e della sua applicazione ai processi di certificazione. In particolare, l'elaborato si inserisce nella gestione delle certificazioni di qualità da parte di un ente certificatore e sull'osservazione dei processi che consentono l'ottenimento dell'attestato. L'utilizzo della blockchain in questo ambito consente l'autosostentamento dell'intero processo di certificazione, permettendo così una migliore organizzazione della rete.

Il tentativo di applicazione della blockchain al contesto delle certificazioni deriva dal crescente interesse nei confronti della nuova tecnologia. Basti pensare al caso Bitcoin oramai fenomeno globale. Nella rete dei Bitcoin i partecipanti possono scambiare criptovalute nonostante sia assente la mutua fiducia. Questo è consentito grazie alla particolare struttura della blockchain che garantisce l'autenticità e veridicità di tutti gli scambi di denaro registrati. Ogni partecipante pone la propria fiducia nell'architettura stessa e nella sua capacità di validare soltanto transazioni consistenti. Ipotizzando quindi l'applicazione della tecnologia anche al contesto delle certificazioni, si potrebbero sviluppare iter certificativi capaci di gestirsi autonomamente.

L'obiettivo dell'elaborato è l'analisi dello sviluppo di una blockchain utile per la gestione degli iter da parte dell'ente certificatore. In particolare, lo studio si concentra sulla generazione di un modello astratto utile per la descrizione di qualunque processo di certificazione. Tale modello risulta fondamentale per l'analisi dei processi certificativi e la loro trasposizione all'interno della blockchain sviluppata su Hyperledger Fabric. Inoltre, al fine di testare la rete così sviluppata, si implementa in Python un emulatore dei processi di certificazione. Lo scopo dell'emulatore è la valutazione delle potenzialità della soluzione proposta e l'individuazione della migliore disposizione nella rete della documentazione utile per l'iter certificativo.

Il lavoro di tesi si articola su cinque capitoli. Nel primo capitolo si definiscono i concetti generali riguardanti la blockchain, cercando di evidenziare le potenzialità della tecnologia e le principali differenze rispetto ai database distribuiti. Il secondo capitolo inizia con una

breve introduzione sul mondo delle certificazioni, descrivendo gli enti in gioco durante il processo e l'utilità del conseguimento dell'attestato. Si passa poi alla definizione del modello astratto generale ipotizzato per la rappresentazione di qualunque processo di certificazione. Nel terzo capitolo l'analisi si concentra sulla descrizione di alcune applicazioni della blockchain ed il suo rapporto contrastante con il GDPR. In esso sono anche ipotizzate tre alternative di posizionamento della documentazione nella struttura della rete (chiamato mapping). Il quarto capitolo si sofferma sulla descrizione dell'emulatore utilizzato durante la fase di test della blockchain. In particolare, si analizzano i requisiti necessari per il suo funzionamento e le caratteristiche degli output ottenuti. Infine, nel quinto ed ultimo capitolo, si procede con l'analisi delle alternative di mapping per individuare la soluzione più vantaggiosa in termini di costi e performance.

Il lavoro di ricerca svolto sulla documentazione per l'ottenimento delle certificazioni ed all'analisi di articoli accademici sul tema della blockchain ha permesso l'impostazione dell'intero studio. Soltanto grazie a questo lavoro, infatti, è stato possibile definire uno modello generale che consenta l'emulazione dell'iter certificativo all'interno della blockchain. Lo sviluppo di quest'unica rete composta da tutti gli enti in gioco durante la certificazione ha reso possibile una gestione ottimizzata dell'intero processo. I risultati derivanti da tale soluzione saranno descritti dettagliatamente nelle conclusioni finali della tesi, evidenziandone vantaggi e svantaggi.

CAPITOLO 1: La tecnologia Blockchain

1.1 La Blockchain

La blockchain è un registro pubblico distribuito nel quale una lista di transazioni è contenuta in una sequenza di blocchi legati tra loro**.** Questa particolare struttura deriva dal contesto operativo in cui si inserisce la tecnologia. La blockchain, infatti, opera in un ambiente in cui è assente la fiducia tra i diversi partecipanti ed il loro rapporto non è mediato da un'autorità superiore. I partecipanti sono individuati nella rete come dei nodi, detti anche *peers*, che si scambiano delle informazioni definite *transazioni*. La blockchain può essere vista analogamente al registro di una banca che amministra le operazioni dei clienti: ogni nuova transazione è aggiunta in fondo al registro (si parla di *append only*) e non è possibile eliminare le transazioni precedenti. A differenza della banca però, non è presente un'autorità che gestisce il registro, ma ognuno dei partecipanti alla rete può agire sul registro stesso [1].

L'obiettivo della blockchain è la gestione di una rete in cui non è presente la fiducia reciproca tra i partecipanti ma è fondamentale la loro partecipazione attiva. Un esempio di tecnologia che utilizza la blockchain come struttura funzionale è Bitcoin, rete in cui i partecipanti si scambiano denaro senza la supervisione di un'autorità superiore. In questo caso le transazioni sono gli spostamenti di criptovalute, chiamate appunto Bitcoin, ed il registro contiene l'elenco di tutti i movimenti di denaro. Nella rete ogni nodo ha timore che gli altri partecipanti si attribuiscano denaro impropriamente e riescano a scrivere sul registro delle transazioni non consistenti. La particolare struttura della blockchain però consente soltanto l'inserimento di transazioni che descrivono uno scambio di criptovalute realmente avvenuto e approvato dalla maggioranza dei nodi. Tutti i nodi possono quindi scambiarsi liberamente denaro, ma soltanto le transazioni che si dimostrano veritiere dopo il controllo della rete vengono poi riportate nel registro.

Il problema della blockchain risulta quindi la mancanza di fiducia tra i partecipanti della rete. Per superare questo limite, ognuno dei nodi possiede una copia del registro che si aggiorna ogni qualvolta una nuova transazione è inserita nel sistema. Rifacendosi all'esempio di Bitcoin, ogni partecipante della rete presenta una copia dell'elenco di tutti gli

spostamenti di criptovalute avvenuti fino a quell'istante. Affinché una nuova transazione sia inserita è fondamentale ottenere il consenso della maggioranza dei nodi attraverso *l'algoritmo del consenso*. Esso è definito nello *smart contract* della blockchain, software in cui sono elencate le principali caratteristiche della rete e punto di incontro tra tutti i partecipanti. Lo smart contract può essere visto come il contratto che gestisce il rapporto tra i nodi, attribuendo ad ognuno di essi un diverso ruolo. Infatti, come riportato nei paragrafi successivi, è possibile che alcuni nodi siano più importanti di altri ed abbiano delle funzioni da svolgere a loro dedicate.

L'algoritmo del consenso è sicuramente una delle parti principali dello smart contract. Esso definisce la metodologia con cui analizzare la veridicità e l'autenticità delle transazioni ed è approvato e riconosciuto da tutti i nodi. I partecipanti, infatti, non presentano fiducia reciproca ma confidano nelle capacità dell'algoritmo di verificare la consistenza delle nuove transazioni. L'algoritmo del consenso risulta quindi il fulcro centrale per raggiungere il consenso distribuito, malgrado la mancanza di mutua fiducia dei nodi. I metodi più utilizzati per ottenere tale consenso, soprattutto nell'ambito delle criptovalute, sono il *Proof-of-Work (PoW)* e *Proof-of-Stake (PoS)*. Nel primo caso, il consenso si ottiene attraverso un processo chiamato *mining* che consiste nell'analisi di particolari funzioni matematiche denominate *funzioni di hash*. Tale meccanismo è quello utilizzato, ad esempio, da Bitcoin e sarà descritto nei paragrafi successivi. Nel Proof-of-Stake, invece, il consenso è ottenuto attraverso l'analisi delle criptovalute effettivamente possedute da ogni singolo partecipante della rete. Questo metodo è utilizzato da Peercoin, altro esempio di criptovaluta, e nasce con l'obiettivo di ridurre i consumi energetici derivanti dal mining [2].

L'analisi dell'autenticità e della veridicità è possibile grazie alla struttura della blockchain stessa. Come mostrato in Figura 1.1, ogni blocco che costituisce la catena è composto da un *block header*, contenente i riferimenti ai blocchi precedenti, ed un *block body* contente le nuove transazioni. Per blocco si intende un insieme di transazioni inserite nella rete in un certo intervallo di tempo. Infatti, durante la definizione della rete stessa, si esplicita la frequenza di inserimento delle transazioni e l'intervallo di tempo entro cui si considerano come unico blocco. Ad esempio, definendo un intervallo di tempo per la generazione dei blocchi pari a 1 secondo ed una frequenza di inserimento di 3 transazioni al secondo, ogni blocco è composto da tre diverse transazioni. Tutte le transazioni, prima di essere inserite nel nuovo blocco, devono superare il consenso dei nodi con uno dei due meccani descritti.

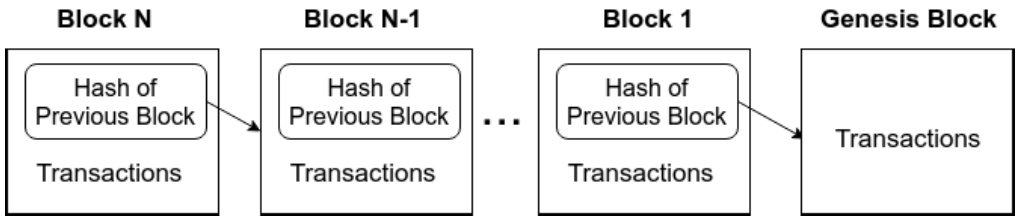

***Figura 1.1**: Struttura della blockchain*

***Fonte**: Tang L, Törngren M., Wang L., A Permissioned Blockchain Based Feature Management System for Assembly Devices, Stoccolma, IEEE, 2020*

L'immagine mostra come sia presente un blocco originario dell'intera struttura, chiamato *genesis block*. Esso è il blocco considerato valido da tutti i partecipanti, senza il quale l'intera struttura perderebbe di significato. Infatti, il primo blocco di transazioni inserito, chiamato *block 1* in figura, è riferito alla *funzione di hash* del contenuto del genesis block. Senza la validità assunta di default del blocco originario, non sarebbe verificabile la consistenza del contenuto del nuovo blocco e di tutti i successivi inseriti nella rete.

Essendo il PoW l'algoritmo del consenso più utilizzato, l'analisi si concentrerà principalmente su di esso e sull'utilizzo delle funzioni di hash. Esse sono delle funzioni matematiche che associano ad un testo di dimensione variabile una stringa di dimensione prefissata. L'utilizzo di tali funzioni è fondamentale per garantire l'immutabilità dei dati all'interno dei singoli blocchi della catena. È possibile, infatti, che la variazione di un singolo carattere del testo originale porti ad un hash finale completamente diverso. Dal punto di vista pratico, la funzione di hash traduce un testo di dimensione qualsiasi in un output di dimensioni prefissate. Un esempio è la funzione "*SHA 256*" che associa ad un testo dato in input una stringa di 256 bit in output. La presenza nel block header della funzione di hash dell'intero blocco precedente garantisce quindi la consistenza delle nuove transazioni. Infatti, anche una piccola modifica del contenuto dei blocchi precedenti genera una funzione di hash differente ed un immediato allarme per tutti i nodi [3].

Grazie alla particolare struttura e agli strumenti di supporto appena descritti, la blockchain presenta alcune caratteristiche peculiari di seguito elencate [4]:

- **Persistenza**: ogni transazione deve ottenere il consenso prima di essere inserita nella blockchain, ostacolando quindi l'inserimento di transazioni non veritiere. Inoltre,

ognuno dei nodi della rete, presenta una copia di tutte le transazioni che complica la modifica delle transazioni già esistenti. Questo comporta l'immutabilità delle transazioni.

- **Decentralizzazione:** la rete generata dalla blockchain è una rete *peer to peer,* ossia una rete paritetica in cui le informazioni possono essere scambiate direttamente tra i nodi. In questa architettura le transazioni tra due nodi sono effettuate senza l'intervento di un'autorità centrale.

- **Anonimità:** i partecipanti interagiscono all'interno della rete utilizzando delle chiavi che garantiscono l'anonimato. In particolare, esistono due tipologie di chiavi: la *chiave pubblica* e la *chiave privata*. La chiave pubblica è l'identificativo del nodo all'interno della rete ed è nota a tutti gli altri nodi. La chiave privata, invece, è nota soltanto al nodo proprietario ed è utilizzata per la firma di ogni transazione inserita del nodo stesso. La chiave pubblica può cambiare ad ogni accesso del nodo, mentre la chiave privata rimane sempre la stessa. Solitamente entrambe le chiavi sono delle stringhe derivanti dalla funzione "SHA-256" vista in precedenza. Ogni volta che si effettua una transazione, il mittente codifica il contenuto della transazione con la sua chiave privata e la invia al destinatario insieme al messaggio non codificato. Il destinatario, attraverso l'utilizzo della chiave pubblica del mittente, decodifica il messaggio e verifica la corrispondenza con il messaggio originale non codificato. Se è presente la corrispondenza, allora la transazione è veritiera. Questa tipologia di codifica prende il nome di criptazione asimmetrica, essendo scrittura e lettura del messaggio eseguiti in istanti diversi.

- **Verificabilità:** ogni blocco contiene tutte le transazioni avvenute in precedenza e tutti i partecipanti alla rete hanno una copia del registro. In ogni momento è possibile quindi effettuare la tracciabilità delle singole transazioni.

Come ultima distinzione, le blockchain si classificano in due grandi famiglie: *permissionless blockchain* e *permissioned blockchain*. Le permissionless, chiamate anche *pubbliche*, sono delle blockchain in cui tutti i nodi hanno la stessa importanza e possono agire sul registro senza restrizioni, rispettando le condizioni dello smart contract. Bitcoin è uno dei maggiori

esponenti della categoria. Le permissioned, chiamate anche *private*, sono delle blockchain in cui alcuni nodi hanno più importanza e possono eseguire delle azioni ad essi riservate: scrittura al registro, approvazione di nuovi nodi etc. In quest'ultime si perde in parte la decentralizzazione per ottenere una maggiore scalabilità e sicurezza. Molto spesso sono blockchain create da enti che desiderano mantenere il controllo della rete, la supervisione degli accessi e del ruolo dei nodi [5].

1.2 Applicazioni della Blockchain: Bitcoin

Negli ultimi anni la blockchain ha riscosso grande successo ed è stata applicata in differenti settori. Il settore alimentare, ad esempio, ha sfruttato la decentralizzazione della blockchain per creare una filiera di rintracciabilità dei prodotti. In questo modo, i clienti possono informarsi sulla storia del prodotto acquistato e individuarne i fornitori. Anche il settore sanitario ne ha sfruttato le potenzialità per lo studio del genoma umano. Grazie ad una collaborazione mondiale di diverse strutture sanitarie è stato possibile creare un'unica blockchain contenente le informazioni sul genoma di milioni di pazienti [6].

Nonostante ciò, il settore pioniere nell'utilizzo della blockchain è stato sicuramente quello finanziario. Nel 2008, Satoshi Nakamoto pubblicò un articolo dal titolo "Bitcoin: A Peer-To-Peer Electronic Cash System" che diede inizio all'era della criptovaluta. Tutt'oggi, dietro alla vera identità dell'ideatore di Bitcoin sono presenti diverse teorie, tra cui la più accreditata è che lo pseudonimo rappresenti un gruppo di ricercatori giapponesi. Bitcoin è una permissionless blockchain nel quale ognuno dei nodi può agire in egual modo sul registro. L'accesso al network è effettuato attraverso un'apposita applicazione scaricata sul telefono o PC. Ognuno dei partecipanti ha un proprio portafoglio chiamato *wallet*, identificato univocamente dalla chiave pubblica, con il quale è possibile scambiare denaro con gli altri nodi della rete.

Lo scambio di denaro è effettuato attraverso il processo schematizzato in Figura 1.2. Un nodo A può inviare dei Bitcoin ad un nodo B spedendo un messaggio contenente i dettagli della transazione. Questa comunicazione è visibile a tutti gli altri nodi della rete ed avviene con il meccanismo di criptazione asimmetrica vista nel paragrafo precedente. Non appena la nuova transazione giunge agli altri nodi della rete, inizia il cosiddetto mining. I nodi, chiamati *miners*, hanno il compito di verificare la validità e veridicità della transazione

attraverso l'analisi delle funzioni di hash. L'obiettivo è la ricerca del collegamento tra il blocco contenente la nuova transazione ed i blocchi già esistenti. Si verifica così se il nodo A possegga effettivamente l'ammontare di bitcoin che ha intenzione di inviare al nodo B, oppure sia soltanto un tentativo di frode.

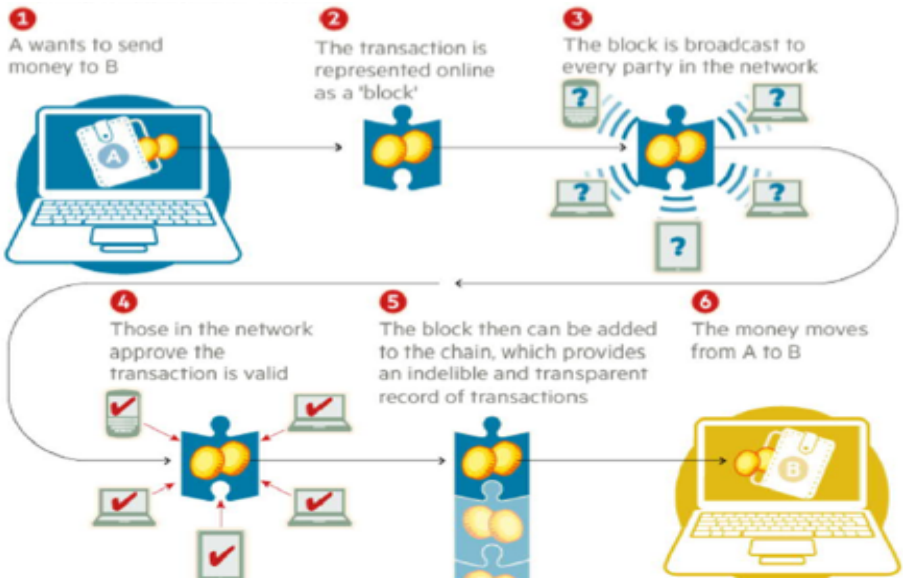

***Figura 1.2**: Funzionamento Bitcoin*

***Fonte**: Crosby M., Nachiappan M., Pattanayac P., Verma S., Kalyanaraman V., Blockchain Technology: Beyond Bitcoin, Berkeley, AIR, 2016, p. 10*

Dal momento che il processo di ricerca richiede un dispendio di tempo ed energie, Bitcoin prevede un rimborso in termini di criptovalute per il primo nodo che riesca a dimostrare la consistenza della transazione. Questo rimborso è proporzionale alla difficoltà della risoluzione del problema matematico e viene ritarato ogni due settimane a seconda del traffico generato in questo periodo di tempo [7].

Risulta evidente come un nodo mal intenzionato possa introdurre nella blockchain, per guadagni personali, delle transazioni inconsistenti. Affinché si verifichi è necessario che il nodo mal intenzionato abbia una potenza di calcolo superiore al 50% della potenza di calcolo di tutti gli altri nodi della rete. Questo perché soltanto così il nodo può essere più veloce degli altri ad approvare la transazione fittizia. Se tale condizione è rispettata, il nodo mal intenzionato ha più probabilità (non la certezza) di riuscire a validare per primo la transazione senza che nessuno si accorga della sua inconsistenza. Una potenza di calcolo

così elevata richiede grandi investimenti che quasi sempre superano i guadagni derivanti da transazioni fittizie. Proprio in questo risiede la capacità di Bitcoin di autosostenersi ed evitare così delle possibili frodi [8].

È importante evidenziare come i Bitcoin si acquistano con valuta reale ma non hanno una valenza monetaria riconosciuta. Ad oggi ancora nessuno stato ha riconosciuto i Bitcoin come moneta con valore intrinseco, nonostante l'Unione Europea stia definendo delle norme per contingentare questo fenomeno mondiale. La più grande preoccupazione deriva dall'intangibilità della criptovaluta che rende fittizio il trasferimento di denaro. Sebbene la valenza non riconosciuta, grandi aziende internazionali hanno effettuato ingenti investimenti in Bitcoin. L'esempio più eclatante è sicuramente quello di Tesla che ha da poco investito 1,5 miliardi di dollari in Bitcoin, accettando anche il pagamento in criptovalute. Questo acquisto ha portato, ad inizio 2021, alla quotazione record di 48mila dollari per singolo Bitcoin [9]. Altre multinazionali stanno valutando lo stesso investimento, aprendo così nuovi orizzonti nel mercato monetario e ad una crescita sempre maggiore del fenomeno Bitcoin.

1.3 Differenze tra Blockchain e Database Distribuito

La possibilità di memorizzare informazioni in modo distribuito suggerisce un'analogia tra blockchain e database distribuito. In realtà, le due strutture presentano grandi differenze già a livello concettuale. Se la blockchain si sviluppa in un ambiente in cui è assente la fiducia tra i diversi nodi, un database distribuito si articola in una rete in cui è presente un'autorità superiore riconosciuta da tutti i partecipanti. Questo porta a scelte progettuali completamente diverse per le due strutture. La blockchain, infatti, necessita di una struttura che consenta la verifica della consistenza delle transazioni. Nei database distribuiti, invece, ogni nuova transazione inserita è valida di default, data la supervisione dell'autorità superiore di cui i nodi hanno fiducia.

Proprio per queste differenze strutturali, le due tecnologie sono utilizzate per esigenze diverse. Il bisogno di performance elevate porta alla scelta di un database distribuito, garantendo maggior scalabilità e minori latenze. La blockchain, invece, è preferibile quando si necessita di elevata sicurezza. In Figura 1.3 è possibile analizzare un confronto sintetico tra le tecnologie. Dal grafico si evidenzia come una permissioned blockchain si inserisca perfettamente come soluzione intermedia tra database distribuito e perminionless

blockchain. Questo perché, come già accennato, le permissioned sono blockchain in cui alcuni nodi hanno più importanza di altri, avvicinandosi molto al concetto di autorità superiore visto nei database. In esse, ad esempio, soltanto determinati nodi possono autenticare la validità delle transazioni e consentirne il loro inserimento.

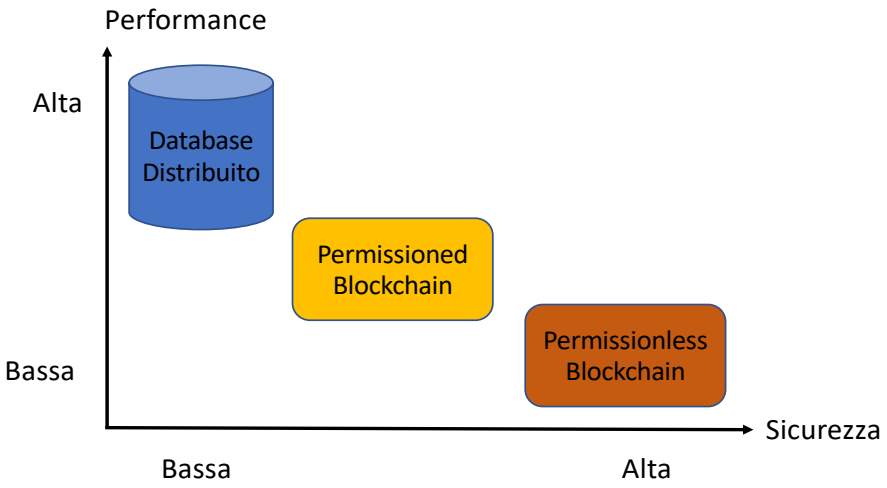

***Figura 1.3**: Confronto tra Blockchain e Database Distribuito*

Questa classificazione deriva da uno studio approfondito delle due strutture su quattro differenti aspetti [10]. Il primo aspetto considerato è quello della *replicabilità*. In una blockchain ogni nodo ha una copia dell'intero registro su cui può effettuare letture o scritture. Tale struttura risulta necessaria nella fase di mining in cui i nodi devono disporre dello storico delle transazioni per validarne una nuova. L'eccessiva replicabilità causa un overhead non trascurabile che riduce le performance, aumentando il traffico nella rete e le latenze. In un database distribuito, invece, ogni nodo può agire soltanto su una porzione di dati alla volta. Il registro ha un'unica copia, oltre a quelle di backup, gestita dall'autorità superiore che stabilisce come i nodi possano agire su di essa. L'autorità rende disponibili i dati secondo ordine e criterio, garantendo quindi migliori performance e soprattutto minori latenze.

Il secondo fattore analizzato è la *contemporaneità*. Per contemporaneità si intende l'inserimento di nuove informazioni all'interno delle due strutture. Nella blockchain l'inserimento di nuove transazioni avviene a seguito della verifica della consistenza, secondo

l'algoritmo del consenso. Questo comporta un'ottica sequenziale in cui è necessario il controllo di una transazione alla volta per garantire una maggiore sicurezza. Nel database distribuito l'inserimento di nuove informazioni avviene in contemporanea seguendo un'ottica in parallelo. Risulta evidente come la gran parte del gap di performance tra blockchain e database risieda proprio nel diverso approccio alla contemporaneità. Il database è in grado schedulare, nella stessa unità di tempo, molte più transazioni rispetto alla blockchain.

Il terzo aspetto è la gestione dello *storage*. Come già detto per la contemporaneità, ogni nodo della blockchain presenta un'intera copia del registro contenente lo storico delle transazioni, necessario per la verifica della consistenza. Tutte le nuove transazioni devono essere replicate su ogni nodo, mantenendo inalterati gli stati precedenti a quello attuale. Nel database distribuito, invece, è presente soltanto l'ultimo stato dell'intero sistema, oltre ad eventuali stati precedenti nel caso in cui si verifichi un crollo improvviso della rete. Questo implica una quantità di memoria superiore per la registrazione di una transazione all'interno della blockchain. Un database distribuito mantenendo soltanto l'ultimo stato del sistema, potenzialmente con una sola copia, risulta meno gravoso dal punto di vista della memoria occupata.

L'ultimo fattore analizzato è l'*atomicità*, riguardante le permissioned blockchain ed i database distribuiti. Essi sono sistemi semi-centralizzati in cui si distribuiscono le diverse parti del registro non essendo presente un unico nodo centrale che si occupa dell'intero storage. In particolare, nei database distribuiti si applica un'ottica per cui ogni nodo possiede quei record su cui agisce maggiormente in lettura o scrittura. I record sono delle porzioni di database, chiamate anche *istanze*, che contengono le informazioni richieste dai nodi attraverso delle *query*. Avvicinando ai nodi le informazioni che richiedono più di frequente, si riducono ulteriormente la latenza ed il traffico sulla rete. Per le permissioned blockchain, invece, si utilizza sempre un'ottica incentrata sulla sicurezza. È possibile ripartire diversamente le copie dei registri a seconda dei nodi, tenendo conto che alcuni nodi con eccessive informazioni possono comportarsi in modo fraudolento.

Le performance più elevate, le minori latenze e l'occupazione ottimizzata dello storage, condurrebbe alla scelta di un database distribuito come soluzione migliore. In realtà, un database distribuito inserito in un ambiente in cui è assente la fiducia tra i nodi perde

completamente le sue potenzialità. Proprio per questo, prima di scegliere quale delle due strutture utilizzare, è fondamentale una profonda conoscenza del contesto in cui si opera. In un ambiente in cui non è presente un'autorità superiore e in cui i nodi non hanno fiducia reciproca, l'unica soluzione che consente di implementare una struttura decentralizzata è la blockchain.

1.4 Hyperledger Fabric

Hyperledger Fabric è un sistema open-source sviluppato dalla Linux Foundation che consente l'implementazione di permissioned blockchain. Il grande vantaggio di questa piattaforma è quello di garantire la personalizzazione dell'intera struttura in via di sviluppo. Nella scrittura dello smart contract, ad esempio, Hyperledger Fabric permette la scelta del linguaggio di programmazione per la sua scrittura, senza la necessità di riferimenti a criptovalute e con la possibilità di modulare i rapporti tra i singoli nodi. Questa elasticità è fondamentale per riuscire a personalizzare la rete in base alle proprie esigenze. Ricordiamo come una permissioned blockchain sia una blockchain privata con una struttura molto simile a quella di un database distribuito. L'ente possessore dell'infrastruttura è interessato a mantenere il controllo della rete e gestire in prima persona il rapporto con i singoli nodi [11].

Hyperledger Fabric per consentire la personalizzazione richiede, oltre alla scrittura dello smart contract, la definizione dell'*endorsement policy*. Se lo smart contract definisce il rapporto tra i nodi della rete, l'endorsement policy definisce l'organigramma dei nodi ed i rispettivi compiti nella stessa. In una blockchain sviluppata su Hyperldeger Fabric, infatti, è fondamentale definire le cosiddette *organisations*, gruppi di nodi in cui è presente la fiducia reciproca. In Figura 1.4 è descritta la struttura di una blockchain sviluppata su Hyperledger Fabric. Nell'esempio sono rappresentate due organizzazioni che partecipano ad una stessa rete. I nodi all'interno delle medesime organizzazioni, individuate in figura come dei blocchi rettangolari, hanno fiducia reciproca e presentano una copia del registro delle transazioni. All'interno di questi blocchi si ha il comportamento tipico di un database distribuito. Nel momento in cui si definisce la rete, chiamata *channel* nell'immagine, non è più presente la fiducia reciproca, essendo essa composta da nodi provenienti da organizzazioni diverse. All'interno del channel si ha un comportamento analogo a quanto visto nella blockchain. Hyperledger Fabric crea quindi una soluzione ibrida: all'interno dell'organizzazione si ha a

che fare con un datatabse distribuito mentre, tra organizzazioni differenti, si ha un comportamento analogo a quello delle blockchain [12].

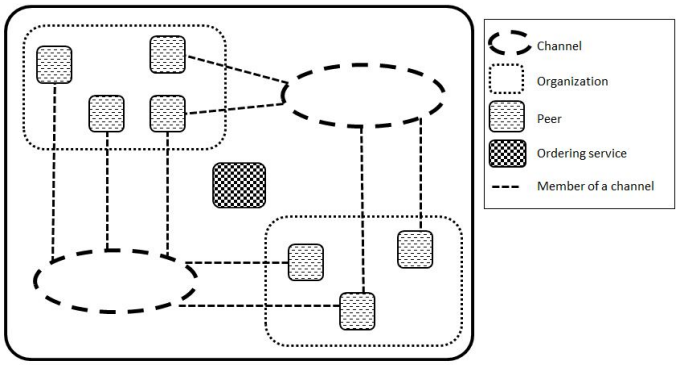

***Figura 1.4**: Struttura Blockchain in Hyperledger Fabric.*
***Fonte**: Jourjon G., Nguyen T.S.L., Potop-Butucaru M., Thai K.L., Impact of network delays on Hyperledger Fabric, Parigi, arXiv, 2019, p.2*

Tale soluzione ibrida conferisce alla blockchain una scalabilità maggiore. Un comportamento tipico dei database all'interno delle organizzazioni, infatti, garantisce la parallelizzazione nell'analisi delle transazioni. Per comprendere meglio i vantaggi derivanti da Hyperledger Fabric, si può osservare il processo di approvazione di una nuova transazione descritto in Figura 1.5.

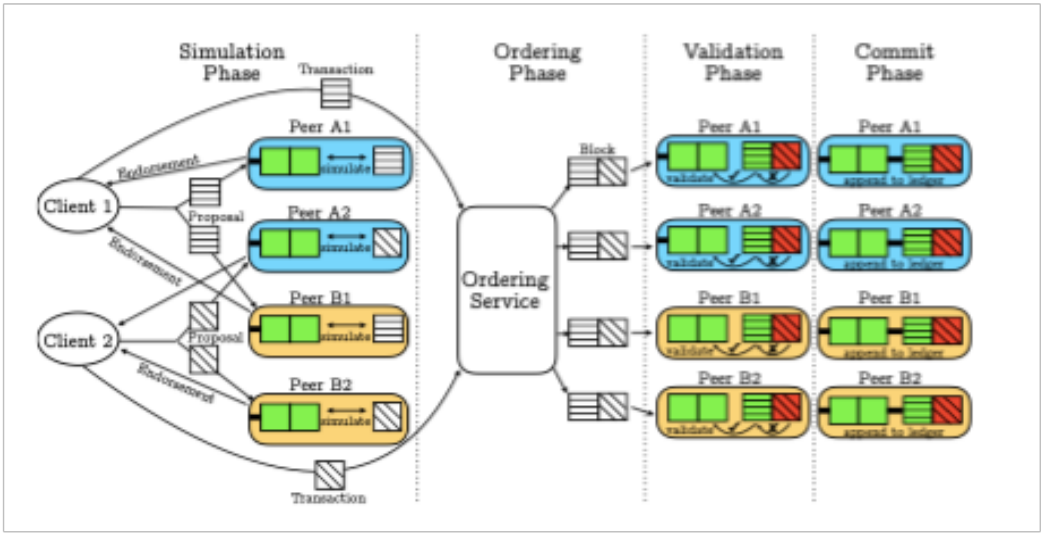

***Figura 1.5**: Funzionamento Hyperledger Fabric*
***Fonte**: Agrawal D., Dittrich J., Shuhuknet F.M., Sharma A., How to Databsify a Blockchain: the Case of Hyperledger Fabric, Saarbruken, arXiv.org, 2018, p.4*

Nell'endorcement policy è fondamentale anche la definizione degli *endorsers*, ossia quei nodi che hanno il compito di verificare la consistenza delle nuove transazioni inserite nella loro organizzazione. Ogni nodo può effettuare una *transaction proposal*, proposta di una transazione con un nodo di un'altra organizzazione. Gli endorsers delle due organizzazioni hanno il compito di analizzare la consistenza della nuova transazione sulla copia locale del loro registro. Se la transazione è consistente su entrambi i registri allora si ha il via libera per il suo inserimento. Questa prima fase è chiamata *simulation phase*. La fase di simulazione ha lo scopo di valutare la veridicità della transazione ed i suoi effetti sul registro nel caso di un eventuale inserimento.

Grazie alla struttura ibrida di Hyperledger Fabric, è possibile effettuare l'analisi di più transaction proposal in parallelo. Per questo è prevista una seconda fase chiamata *ordering phase*, nel quale le transazioni che hanno ottenuto l'*endorsment*, ossia l'approvazione degli endorsers, sono ordinate. Solitamente l'approccio utilizzato è quello che in logistica prende il nome di *First In First Out (FIFO)*: le transazioni che ottengono prima l'approvazione risultano processate per prime. L'ordering phase ha anche lo scopo di raggruppare in blocchi le transazioni simili, riducendo così il traffico nella rete. I nodi della rete che si occupano di tale schedulazione prendono il nome di *orderers*

I blocchi di transazioni così ordinati sono inviati a tutti i nodi per la *validation phase*. Questa, a differenza della simulation phase, prevede un controllo di consistenza svolto da tutti i nodi e non soltanto dagli endorsers. Ovviamente, l'analisi di consistenza presume lo studio delle funzioni di hash per collegare le nuove transazioni ai blocchi già esistenti e verificarne così l'autenticità e veridicità. Soltanto i blocchi per cui l'hash è ricondotto a blocchi già esistente possono ottenere la validazione.

Una volta approvata dall'intera rete, la transazione è soggetta alla *commit phase*. Essa è la quarta ed ultima fase ed ha il compito di legare il blocco di transazioni approvato alla copia in comune del registro. Così facendo il registro è aggiornato all'ultimo stato del sistema contenente anche le nuove transazioni appena introdotte [13].

Nel progetto si è deciso di utilizzare Hyperledger Fabric come ambiente di sviluppo della permissioned blockchain. La scelta deriva dagli evidenti benefici nell'utilizzo della piattaforma: riduzione delle latenze, maggiori performance, ottimizzazione dello storage e

riduzione del traffico. Nonostante i vantaggi, la piattaforma non è ancora completamente matura e non di rado si verifica la scomparsa improvvisa di transazioni dalla rete. Questo implica grandi difficoltà nell'analisi della consistenza, aumentando le latenze nel caso in cui non sia più presente il blocco di riferimento della transazione. Inoltre, la scomparsa del blocco di confronto dalla rete può portare al mancato inserimento di nuove transazioni anche se realmente consistenti.

Oltre alla scomparsa di transazioni, come mostrato in Figura 1.6, le performance di Hyperledger Fabric dipendono anche dalla grandezza dei singoli blocchi e dalle dimensioni del registro. Nel dettaglio, si valuta la velocità con cui la rete è in grado di verificare l'autenticità delle transazioni. Più il numero di blocchi inseriti è grande, più tempo è richiesto dalla rete per l'analisi della consistenza. In figura si può notare come un registro costituito da 10000 blocchi impieghi quasi il doppio del tempo per ottenere il consenso rispetto ad un registro composto da 5000 unità. Inoltre, a parità di dimensioni, il tempo per la verifica aumenta con la grandezza del blocco, ossia con il numero di transazioni che lo compongono. Ad esempio, se si considera un registro di dimensione 1000, i blocchi composti da un numero minore di transazioni sono validati più velocemente [14].

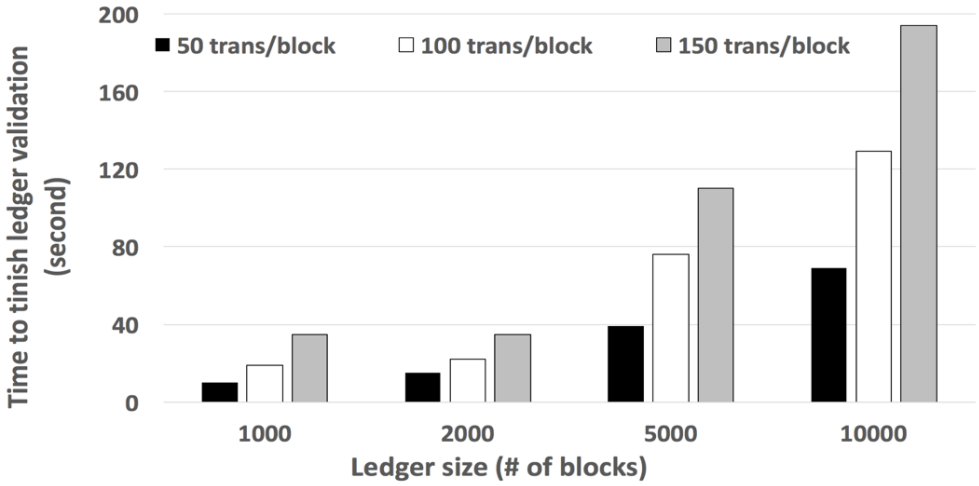

***Figura 1.6**: Velocità di validazione in Hyperledger Fabric*

***Fonte**: Barger A., Baset S., Dillemberg D., Manevich Y., Novotny P., Zhang Q., LedgerGuard: Improving Blockchain Ledger Dependability, Yorktown, arXiv.org, 2018, p.6*

L'analisi delle potenzialità e dei limiti di Hyperledger Fabric ha consentito di scegliere le alternative migliori per lo sviluppo della blockchain sulla piattaforma. Nei prossimi capitoli

si cercherà di mettere in evidenza le modalità di implementazione della rete. In particolare, come sia stato possibile conciliare la tecnologia blockchain con i processi di certificazione, partendo proprio dallo studio di tali processi e dallo sviluppo di un modello generale valido per tutte le certificazioni.

CAPITOLO 2: Il Modello delle certificazioni

2.1 Il processo di certificazione

Da sempre il concetto di qualità ha influenzato l'ambito produttivo. La ricerca del miglioramento continuo, raggiungibile attraverso l'analisi e la riduzione degli sprechi, ha permesso una variazione nella visione di questa funzione aziendale.

La qualità come la conosciamo oggi deriva da un lungo processo evolutivo che dura da più di un secolo. Inizialmente per qualità si intendeva il controllo dei prodotti finiti, fonte di costi, rallentamenti e occupazione delle risorse. Spesso il controllo finale dei prodotti risultava il vero collo di bottiglia dell'intero processo produttivo. Sotto quest'ottica la qualità appariva come un costo, legata indissolubilmente ai prodotti. Con il passare degli anni si è abbandonato il concetto di controllo di qualità a favore di un sistema per la gestione della stessa. Lo scopo non era solo più il controllo della qualità ma la sua pianificazione. Essa è così divenuta un metodo per gestire l'organizzazione, trasformandosi da costo ad investimento e slegandosi dal solo ambito dei prodotti. Iniziano così a svilupparsi le prime organizzazioni che cercano di applicare i principi della qualità anche all'ambito dei servizi.

Un'impresa che opera nell'ottica della qualità ha una razionalizzazione del sistema organizzativo che incentiva il miglioramento continuo, il raggiungimento degli obiettivi aziendali ed il soddisfacimento dei clienti. Questo comporta alti livelli prestazionali garantiti anche dall'utilizzo di appositi strumenti di progettazione come il *Quality Function Deployment (QFD)*, le tecniche di *Computer Aided (CAx)* e le tecniche *Design For (DFx)*. Tali tecniche sono utilizzate nella fase di progettazione per prevenire gli eventuali difetti. È importante sottolineare come l'inserimento della qualità a monte del processo produttivo non abbia comunque eliminato i controlli sugli output. La progettazione della qualità, infatti, deve verificarsi anche nella produzione. Proprio per questo si sono sviluppate delle tecniche per il controllo statistico della produzione come le carte di controllo ed i piani di campionamento. Il controllo, chiamato *Statistical Process Control (SPC)*, prevede un controllo campionario della produzione ed un'analisi della qualità per ogni campione estratto. [15].

Risulta chiara l'attenzione delle imprese a mostrarsi come organizzazioni operanti nell'ottica della qualità. I clienti si interessano alle organizzazioni che riescono a distinguersi nel mercato come le più attente alle esigenze dei consumatori. Il problema è la possibilità che ha ogni impresa di auto-definirsi operante sotto i principi della qualità, facendo perdere di significato a questo concetto. Proprio a tale scopo sono stati definiti degli standard internazionali entro cui una determinata impresa debba operare per ottenere delle certificazioni di qualità. Questi sono contenuti in norme emanate dall'*International Standard Organisation (ISO)*, ente internazionale che si occupa della definizione di standard in ambito produttivo. La famiglia di norme che definisce gli standard per qualificare un sistema di gestione per la qualità è la serie 9000. In particolare, la norma ISO 9000 definisci i concetti di qualità mentre la ISO 9001 evidenzia le caratteristiche di un sistema di gestione della qualità. In queste norme sono rimarcati alcuni aspetti fondamentali per un'organizzazione che opera nell'ottica della qualità, tra cui leadership, coinvolgimento del personale, approccio sistematico alla gestione, sempre con l'obiettivo del miglioramento continuo [16].

Si definisce quindi un processo per verificare la capacità dell'organizzazione di operare secondo i principi definiti dalle norme. Tale processo prende il nome di certificazione. In Figura 2.1 sono individuati i principali attori che partecipano all'iter certificativo.

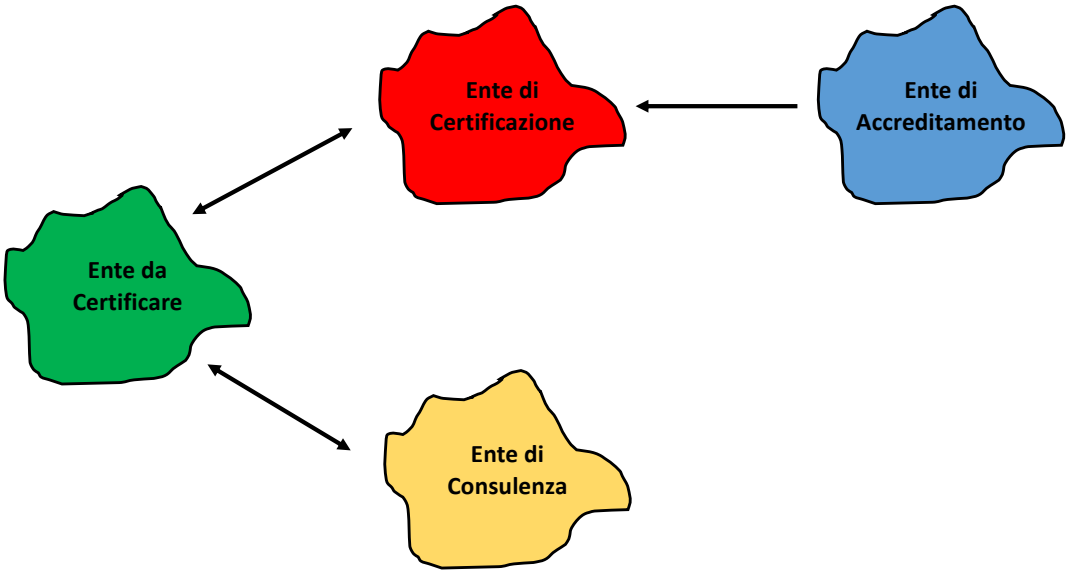

***Figura 2.1**: Entità coinvolte nel processo di certificazione*

Lo schema è valido per tutte le certificazioni, da quelle di un singolo prodotto a quelle di un intero processo. Nel dettaglio si analizza il caso di una certificazione ISO 9001, in cui l'impresa vuole essere riconosciuta come organizzazione che opera seguendo i principi della qualità. La certificazione ISO 9001 non è obbligatoria ma è strettamente necessaria per tutte quelle imprese che desiderano essere competitive sul mercato. Possedere tale certificazione, infatti, consente di testimoniare a tutti gli stakeholders dell'impresa (fornitori, investitori, clienti etc.) l'effettiva qualità dell'organizzazione. Questo permette di ridurre i costi di negoziazione derivanti dalla distanza tra le parti sull'effettiva qualità del prodotto o servizio fornito dall'impresa. In un mercato competitivo minori costi equivalgono a maggiori investimenti, ossia maggiori possibilità di creare un vantaggio competitivo.

La richiesta di certificazione è fatta dall'organizzazione che vuole certificarsi, chiamata in questo contesto *ente secondo*. Quando l'organizzazione crede di rispettare tutte le direttive previste della norma contatta un ente di certificazione, chiamato *ente terzo*, per organizzare il cosiddetto *audit*. In questo caso, trattandosi di un ente esterno che esegue il controllo, si parla di *audit esterno*. Esistono anche gli *audit interni*, eseguiti da esperti della qualità interni all'impresa, cha hanno lo scopo di verificare la corrispondenza tra qualità progettata e realizzata. Molto spesso gli audit interni sono propedeutici per gli audit esterni verificando anticipatamente la presenza di non conformità alle norme. L'ente di certificazione, ricevuta la richiesta, invia un preventivo all'organizzazione contenente il prezzo dell'intero processo di certificazione. Questo prezzo tiene conto principalmente delle giornate di lavoro, del tipo di certificazione e del numero di auditor impiegati durante il controllo. Se l'organizzazione accetta il preventivo, si passa alla definizione di una data per il controllo sul campo.

L'ente di certificazione può eseguire le ispezioni poiché è accreditato dall'*ente di accreditamento*, ente unico nazionale designato dal governo. In Italia l'ente di accreditamento è Accredia. Il compito di Accredia è quello di definire gli enti di certificazione sul territorio italiano e, per ognuno di questi, indicare quali certificati possano conferire alle singole organizzazioni. La presenza di un ente di supervisione dell'intero mercato delle certificazioni fa sì che queste abbiano una valenza riconosciuta da tutti [17].

Nel caso di imprese di piccole dimensioni è possibile che esse non abbiano le capacità di preparare tutta la documentazione necessaria per richiedere la certificazione. Non avendo del personale interno addetto all'ambito della qualità, si possono rivolgere ad una società di

consulenza. Essa, sotto corrispettivo economico, prepara l'impresa per l'audit facendo un'analisi preliminare della qualità interna all'organizzazione. Se ritiene l'impresa idonea all'ottenimento della certificazione, la coadiuva alla preparazione della richiesta ed alla raccolta di tutta la documentazione necessaria.

Nella data prefissata per l'audit, gli auditor provvedono al controllo della conformità dell'impresa alle norme. Il controllo può portare a tre esiti differenti:

- **Conforme**: l'impresa è conforme ed è pronta a ricevere la certificazione

- **Non conformità lievi**: l'impresa presenta delle non conformità alle norme che possono essere facilmente corrette. Si organizza un nuovo audit nel breve periodo.

- **Non conformità gravi**: l'impresa presenta grosse discrepanze rispetto alla norma di riferimento. Si organizza un nuovo audit nel lungo periodo.

Se l'organizzazione riesce ad ottenere la certificazione, questa ha una valenza di tre anni. Ogni anno l'ente di certificazione esegue degli audit intermedi, chiamati *audit di sorveglianza*, per verificare il mantenimento delle conformità alle norme. Gli audit intermedi sono meno intrusivi rispetto agli audit esterni veri e propri e risultano già conteggiati nel primo preventivo consegnato all'impresa stessa [18].

L'organizzazione può anche procedere con un'autocertificazione della qualità. Attraverso audit interni ha la possibilità di dimostrare a terzi il rispetto delle linee guida. In questo caso si parla assicurazione della qualità (*assurance of quality*). Per dimostrare il perseguimento della qualità è necessaria la presenza del *manuale della qualità*, registro in cui si raccoglie la documentazione dei processi produttivi e si definiscono le modalità di controllo degli stessi. Il manuale funge anche da guida durante gli audit interni per la verifica della qualità: è il riferimento con cui confrontare l'andamento effettivo dei processi. Ovviamente, l'autocertificazione ha una valenza minore della certificazione vera e propria. In quest'ultima, infatti, è presente un ente terzo accreditato dall'ente unico di accreditamento che garantisce a tutti come l'impresa rispetti fedelmente le norme.

2.2 Il modello astratto

Il paragrafo precedente ha introdotto il concetto di certificazione e le sue implicazioni sul mercato. Questo studio è stato fondamentale per la costruzione di un modello valido per tutti i processi di controllo, che definisse uno schema generale applicabile ad una qualsiasi certificazione. Il modello si concentra maggiormente sulla fase di audit vera e propria, in cui gli attori presenti sono l'ente di certificazione e l'organizzazione che vuole certificarsi.

L'idea nasce dallo studio nel dettaglio di un processo di certificazione per la norma ISO 9001. In Figura 2.2 è rappresentato lo schema del modello ipotizzato.

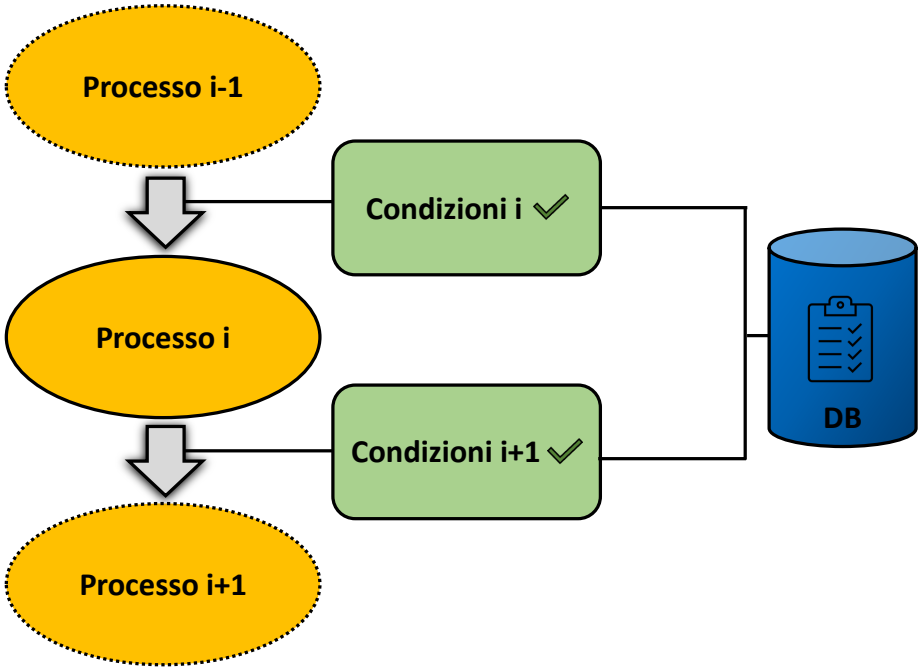

***Figura 2.2**: Modello astratto delle certificazioni*

Il raggiungimento dell'attestato di certificazione è una sequenza di stadi successivi che si realizzano soltanto al verificarsi di determinate condizioni. Queste condizioni sono l'invio o la ricezione di alcuni documenti compilati da parte di uno degli attori in gioco. Ciò significa che, per passare dallo stadio i allo stadio i+1 (chiamati processo i e processo i+1 poiché sono delle vere e proprie azioni da svolgere), è necessaria la presenza di documentazione firmata e controfirmata. Tale documentazione è salvata su un database solitamente esterno all'ente di certificazione (DB in figura). Sarà poi obiettivo dei capitoli successivi comprendere se sia

più opportuno l'utilizzo di un database esterno all'ente di certificazione oppure prevedere l'introduzione diretta dei documenti all'interno della blockchain.

Ovviamente, se le condizioni non risultano verificate, il processo di certificazione termina con un insuccesso. Quasi sempre il fallimento del processo di certificazione deriva dall'eccessivo tempo trascorso tra la richiesta e l'eventuale risposta per la documentazione. Infatti, si ipotizza che il passaggio da un processo ad un altro abbia dei limiti in termine di date. Se la compilazione del documento non avviene entro la data prestabilita e se, anche i solleciti risultano ignorati, il processo di certificazione termina con un insuccesso. È importante evidenziare come un eventuale blocco del processo è possibile in uno qualunque dei processi che compongono l'iter. Nel modello sarebbe opportuno inserire un'ulteriore freccia di uscita da ogni processo che conduce alla conclusione dell'iter di certificazione.

Il modello è sicuramente molto schematico e non comprende alcuni fattori che molto spesso intervengono nelle certificazioni. L'obiettivo non era la definizione di un modello completo che descrivesse tutti gli aspetti della certificazione, ma uno scheletro applicabile a tutti i processi. Per comprendere al meglio la sua efficacia, nel paragrafo successivo è proposta una sua applicazione al processo di certificazione della norma ISO 9001. Si vuole così dimostrare come il modello sia applicabile a qualsiasi tipologia di certificazione, anche a quelle più complesse come la ISO 9001.

2.3 Applicazione del modello alla certificazione ISO 9001

Si è già visto come la norma ISO 9001 riguardi la certificazione di un sistema di gestione per la qualità (detto anche sistema di qualità). Il processo per ottenere tale certificazione richiede l'utilizzo di molte risorse, sia fisiche che di tempo. A pagina 25, in Figura 1.3, è schematizzato il processo di certificazione per la norma ISO 9001 seguendo il modello introdotto nel paragrafo precedente. Per semplicità di rappresentazione è considerato soltanto il caso "positivo", ossia quando il processo termina con il conseguimento della certificazione. Non si è tenuto conto degli audit di sorveglianza che si effettuano nei due anni successivi al conseguimento della certificazione (essendo una ripetizione del medesimo processo). Inoltre, come spiegato nel modello stesso, ogni documento è in realtà memorizzato in un database che non è rappresentato nello schema.

La prima fase del processo è la compilazione del documento di richiesta d'offerta per la certificazione. Il documento è scaricato direttamente dal sito dell'ente certificatore e compilato dall'organizzazione che intende ottenere la certificazione. La compilazione prevede l'introduzione di alcune informazioni generali dell'organizzazione: tipologia di certificazione richiesta, settore, numero dipendenti etc. In questa prima fase l'organizzazione completa un questionario informativo nel quale descrive le sue principali caratteristiche, mettendo in evidenza il codice ATECO corrispondente alla sua attività economica. La definizione del codice ATECO è fondamentale poiché ad esso sono associati dei codici EA derivanti da accordi stipulati tra gli organi europei di accreditamento. Per uniformare a livello comunitario le certificazioni, infatti, gli enti di accreditamento come Accredia hanno firmato un accordo multilaterale nella definizione dei settori d'impresa. Tale accordo prende il nome di EA MLA (EA Multilateral Agreement). Così facendo un'organizzazione che ottiene la certificazione da un ente accreditato per il settore in cui opera l'impresa stessa, può far valere il certificato in tutti gli stati che hanno firmato l'accordo.

Nel momento in cui l'ente riceve la documentazione, controlla la correttezza della stessa e, attraverso le tabelle di calcolo delle giornate-uomo, effettua la sua offerta per l'intero processo di certificazione. L'ente di certificazione che riceve la domanda deve essere necessariamente accreditato con il codice EA corrispondente al codice ATECO presentato dall'organizzazione. Se così non fosse, l'eventuale certificato emesso sarebbe nullo in partenza. Il calcolo dell'offerta dipende principalmente dalla grandezza e natura dell'organizzazione, dal tipo di certificazione e dal numero di giornate e auditor necessari per il controllo.

A questo punto l'organizzazione valuta l'offerta in tutti i suoi termini. Se l'offerta è accettata, il responsabile tecnico dell'ente di certificazione individua il team di auditor più opportuno per la verifica sul campo (attraverso opportune tabelle documentate). Gli auditor selezionati devono possedere le competenze tecniche dell'ambito indicato nel codice ATECO dell'organizzazione. Al team selezionato è inviata la lettera di incarico e, in contemporanea, la segreteria tecnica prepara la documentazione per la pianificazione degli incontri e la definizione di possibili date per i controlli.

Per continuare la procedura è necessario attendere la conferma di auditor e organizzazione. Nel momento in cui entrambi confermano la loro disponibilità, inizia l'audit vero e proprio.

In questa fase vi è il controllo sul campo della bontà del sistema di gestione dell'organizzazione. Gli auditor verificano, tra le altre cose, se quanto dichiarato dall'organizzazione nel manuale della qualità sia effettivamente realizzato. Il manuale della qualità, come già anticipato, è un manuale contenente tutta la documentazione riguardante le modalità di controllo della qualità all'interno dell'impresa. Al termine dell'audit, diviso in fase I e fase II, l'auditor riassume la visita in un report finale. È possibile che, prima del report finale, l'auditor consigli opportune modifiche per correggere eventuali non conformità più o meno gravi riscontrare (riportate tutte in un'opportuna documentazione).

Il report viene poi consegnato al responsabile tecnico dell'ente di certificazione che analizza nei dettagli la pratica. Se non sono presenti discrepanze, il responsabile tecnico prepara il documento di riesamina da presentare al comitato di certificazione. Il comitato è l'organo che si occupa di ricontrollare tutto il processo di certificazione e di emettere il certificato stesso. Esso è composto da esperti tecnici del settore in cui opera l'organizzazione. Se il comitato conferma la correttezza dell'iter, il processo termina con la consegna all'organizzazione del certificato di conformità alla normativa ISO 9001.

La figura riassume sinteticamente quanto descritto. Come già accennato, molti aspetti sono stati trascurati per favorire una più chiara lettura e standardizzazione del modello. La definizione di questo modello è stata fondamentale per comprendere la modalità di inserimento delle transazioni all'interno della blockchain. Nel successivo capitolo si cercherà di capire come sia più opportuno conciliare tale modello con la struttura distribuita della blockchain. Si discuteranno nel dettaglio tre possibili approcci di interazione tra processo di certificazione, schedulazione e memorizzazione delle transazioni inserite nella blockchain.

Legenda

O: Organizzazione che richiede la certificazione

E: Ente di certificazione

A: Auditor

: Transizione da un processo ad un altro

: Processo

: Condizioni verificate grazie alla compilazione/firma del documento

: Inizio/fine del processo

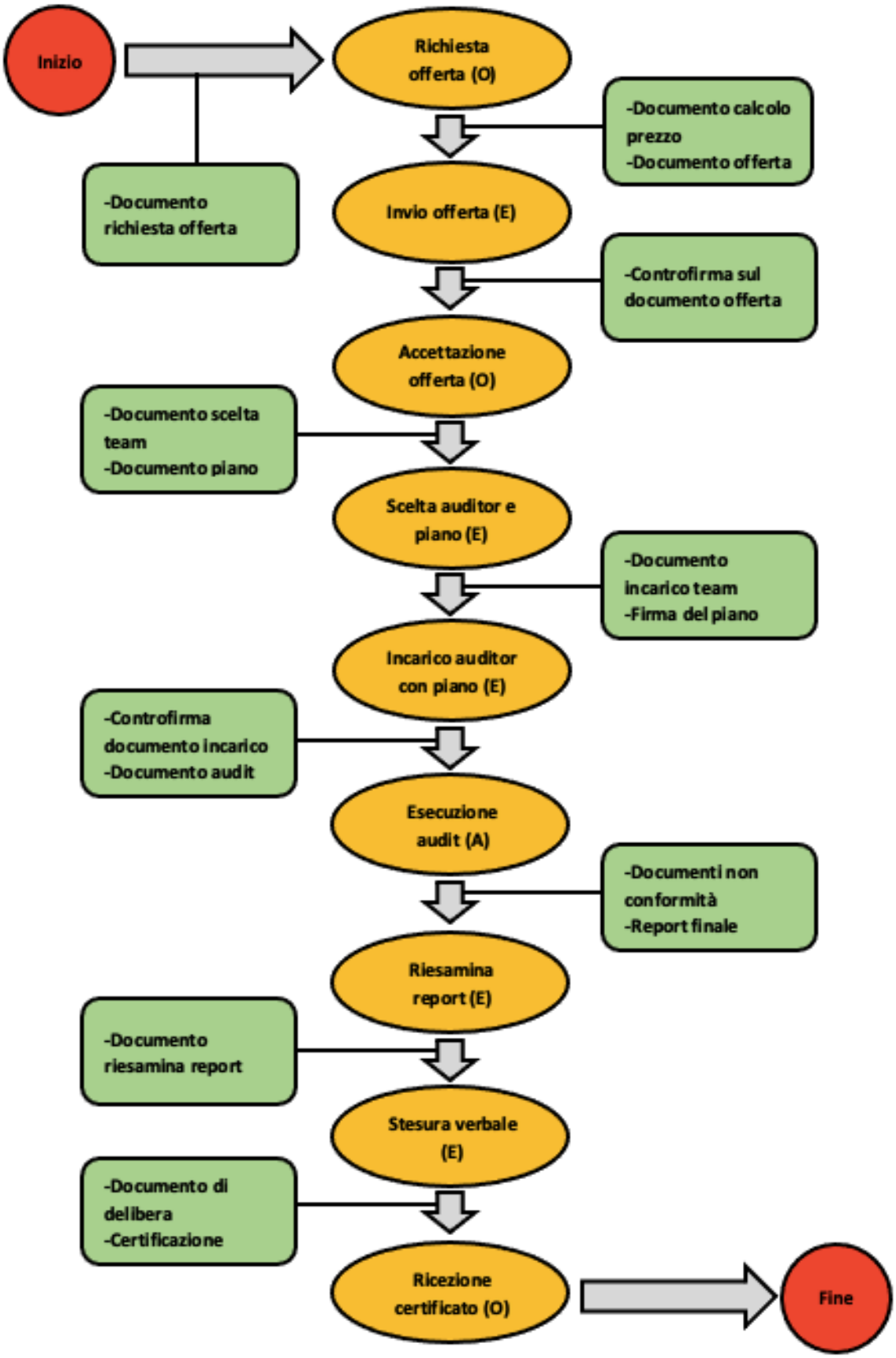

***Figura 2.3**: Modello applicato alla certificazione ISO 9001*

CAPITOLO 3: Mapping nella Blockchain

3.1 Applicazione della Blockchain nel contesto della qualità

L'individuazione di una struttura che coniugasse il modello astratto appena descritto con la blockchain, è cominciata con l'analisi di architetture già utilizzanti la tecnologia come strumento per la qualità. Di seguito saranno presentati alcuni esempi di blockchain applicate come metodo di monitoraggio all'interno del contesto produttivo e non solo.

Un primo esempio è il tracciamento dei prodotti all'interno della filiera produttiva, già adoperato tutt'oggi da molte imprese. L'utilizzo di una blockchain per la raccolta di tutti i dati relativi ad un singolo prodotto, infatti, consente l'individuazione dell'istante in cui iniziano a mancare alcuni aspetti della qualità. L'analisi dei dati riesce a fornire una giustificazione sul motivo per cui il prodotto non rispetti i vincoli imposti nella progettazione: macchina difettosa, materia prima non adatta, operatore disattento etc. In Figura 3.1 è rappresentata la struttura della rete utilizzata in questo tipo di blockchain.

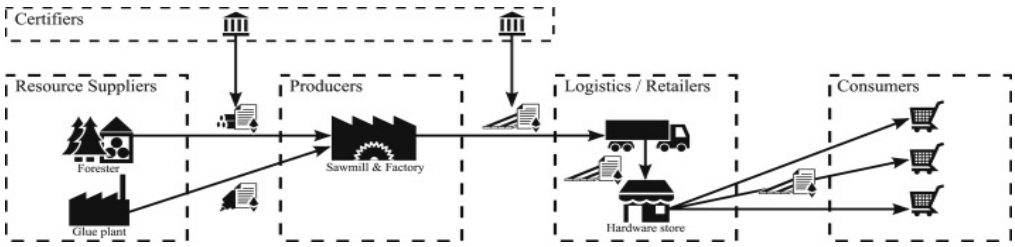

***Figura 3.1**: Struttura della rete in una blockchain nella supply chain.*
***Fonte**: Küpper A., Victor F., Westerkamp M., Tracing manufacturin process using blockchain-based token compositions, Berlino, Digital Communications and Networks, 2020*

Nella rete sono presenti: l'impresa produttiva, i clienti ed i fornitori. L'introduzione nella blockchain di tutti gli attori della filiera rende disponibile il tracciamento e lo storico dei prodotti a tutti gli stakeholders. Inoltre, ogni transazione è certificata da un'autorità superiore che quasi sempre coincide con l'impresa produttiva. L'utilizzo di tale blockchain consente ai clienti di avere a disposizione lo storico del prodotto e di individuare il momento in cui viene a mancare la qualità. Inoltre, la presenza dei fornitori comporta dei vantaggi anche alle

imprese produttrici. L'impresa, attraverso l'analisi delle transazioni passate, è in grado di analizzare l'affidabilità dei fornitori e stipulare dei contratti personalizzati a seconda della fiducia nei confronti degli stessi [19].

Ciò consente un miglioramento del Quality Management System (QSM) delle imprese. Il continuo tracciamento dei prodotti dall'approvvigionamento della materia prima fino alla consegna dei prodotti finiti al cliente, permette un monitoraggio continuo di tutta la filiera. Così facendo, l'individuazione dei difetti è molto più veloce e non è necessario sostenere costi di negoziazione troppo elevati. I clienti, infatti, conoscono lo storico dei prodotti e l'impresa conosce lo storico dei fornitori. La creazione di un ecosistema per il tracciamento dei prodotti contenente tutti gli attori principali della filiera, consente un grande risparmio sui costi della qualità, composti principalmente dai costi di negoziazione [20].

L'applicazione della blockchain per il monitoraggio della qualità ha consentito di prevenire possibili plagi nell'ambito produttivo. Negli ultimi decenni, lo sviluppo della tecnologia delle stampanti 3D è cresciuto esponenzialmente. Questo ha velocizzato il processo di produzione ed esposto le imprese ad un rischio di plagio sempre più evdiente. Molto spesso, infatti, le copie riescono ad arrivare sul mercato prima ancora dei prodotti originali. L'utilizzo della blockchain in questo contesto garantisce un'efficace prevenzione dal plagio grazie all'inserimento nella rete dei contratti di licenza. In Figura 3.1 è rappresentata la struttura del sistema utilizzato nell'ambito della produzione con stampanti 3D.

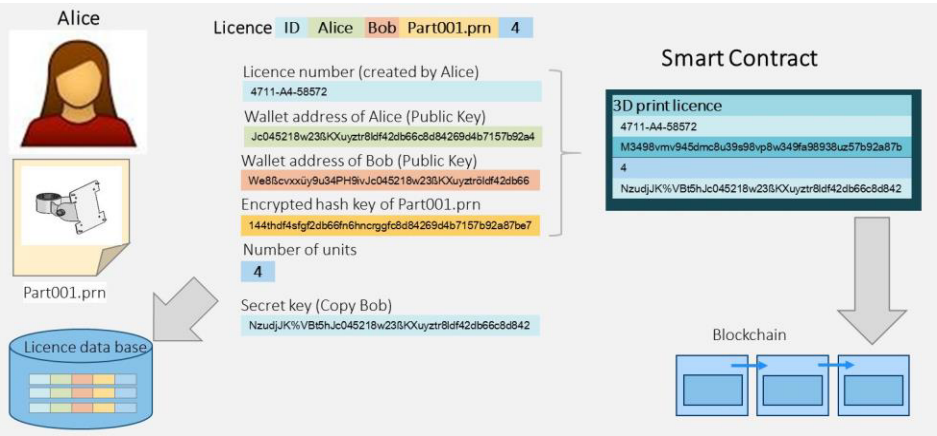

***Figura 3.2**: Sistema integrato per la stampa 3D*

***Fonte**: Nigischer C., Stjepandic J., Copyright Protection in Additive Manufacturing with Blockchain Approach, Singapore, IOS Press, p. 918*

L'impresa che possiede i brevetti del prodotto può conferirli in licenza per consentirne la produzione ad altre imprese. Ovviamente, la licenza di produzione ha un costo ed è vincolata da contratti che si riferiscono alle norme sulla violazione del copyright. Questi contratti devono essere inseriti all'interno della blockchain affinché si possa verificare la consistenza delle transazioni. In figura si evidenzia come la blockchain sia collegata ad un database esterno contenente tali contratti di licenza. Ogni qualvolta ad un nodo della rete è richiesta la produzione di un certo numero di pezzi, si crea una nuova transazione da inserire nella blockchain. Prima che la produzione possa partire, la transazione deve ottenere l'approvazione del network. Lo smart contract del sistema prevede che le stampanti 3D di tutti i nodi della rete possano stampare soltanto quando la transazione riceve l'approvazione. In questo modo si crea un network in grado di autosostenersi e che non permette la stampa dei prodotti a chi non possiede la licenza. L'intero sistema prende il nome di SAMPL (Secure Additive Manufacturing Platform) ed ha l'obiettivo di riunire tutte le imprese operanti nello stesso campo per un controllo più completo delle violazioni dei brevetti. Un sistema di questo genere ha un grande potenziale e consentirebbe di risparmiare tempo e denaro nelle cause per violazione dei diritti di produzione. Le imprese lavorerebbero in sintonia, senza la necessità di singole azioni legali [21].

Un ultimo esempio è l'utilizzo della blockchain al contesto universitario. Uno studio di Hong Kong ha tentato di classificare le università dello stato definendo un processo analogo a quello di certificazione di un sistema di qualità [22]. Il riferimento utilizzato per il metodo è la norma ISO 9001 già affrontata. L'idea alla base del modello è la definizione di processi standard per definire la qualità all'interno del contesto universitario. La grande differenza con i sistemi produttivi risiede nella intangibilità dell'istruzione. L'istruzione è un servizio, non un prodotto, di conseguenza non è presente la componente tangibile che facilita l'analisi della qualità. Proprio per questo lo studio ha definito alcune dimensioni caratteristiche per la descrizione della qualità all'interno dell'ambito universitario, rifacendosi al modello SERVQUAL tipico dei servizi [23]. Questo modello presenta cinque dimensioni caratteristiche:

- **Elementi tangibili**: strutture fisiche, attrezzature e personale messo a disposizione ed utilizzato nell'erogazione del servizio.

- **Capacità di risposta**: volontà nel fornire il servizio con prontezza e garantire aiuto ai clienti quando richiesto.

- **Capacità di rassicurazione**: competenza, cortesia e credibilità degli addetti ai lavori.

- **Empatia**: capacità di immedesimarsi nel cliente, riuscendo a comprenderne le sue esigenze.

- **Affidabilità**: erogazione del servizio in modo affidabile e preciso.

Tutti i servizi possono essere valutati su queste dimensioni, cercando di misurare come il servizio riesca a soddisfare ognuna di esse. Molto spesso il loro peso è diverso sulla base del contesto che si sta valutando.

Le università di Hong Kong sono state analizzate su queste cinque dimensioni e le loro valutazioni inserite all'interno di una blockchain. Tale network ha come nodi le università stesse, e ha permesso di definire una sorta di benchmarking per il livello di istruzione. Un'università che raggiunge alti valori su tutte le dimensioni è un buon esempio da seguire per tutte le altre. Inoltre, avendo a disposizione tutte le informazioni, l'iter di valutazione delle dimensioni e le valutazioni stesse, è possibile capire in che modo sia possibile il miglioramento per ognuna delle università. Ad esempio, un'università carente nell'aspetto empatico deve comprendere le motivazioni di questa carenza. Il miglioramento inizia prendendo come riferimento i processi utilizzati da un'università leader in questa dimensione.

Questi esempi hanno dimostrato come, allo stato attuale delle cose, la blockchain sia utilizzata principalmente come strumento per il monitoraggio della filiera. Inoltre, il suo utilizzo è spesso associato ad un database esterno o alla possibilità di memorizzare i dati nella blockchain stessa. La struttura distribuita richiede l'utilizzo massivo dei dati raccolti per garantire il consenso in un ambiente in cui è assente la fiducia reciproca tra i partecipanti. Su queste considerazioni sono idealizzate le tre alternative esposte nei paragrafi successivi.

3.2 Blockchain e GDPR

Prima di procedere con l'analisi dei tre modelli ipotizzati, è importante soffermarsi su un tema molto sentito in questi ultimi anni: la privacy. In particolare, con l'avvento dei social, l'attenzione alla protezione dei dati personali è sempre più evidente. Tutte le comunità mondiali stanno procedendo con una regolamentazione serrata nel campo del trattamento dei dati sensibili. Questo garantisce una maggiore protezione degli utenti ma, dall'altra parte, limita il potenziale nell'implementazione di molte piattaforme. La blockchain, ad esempio, funziona grazie alla raccolta e immagazzinamento dei dati. Tutti i partecipanti della rete, infatti, presentano potenzialmente una copia del registro contenente tutte le transazioni passate di ogni nodo del network. Una rigida regolamentazione porterebbe alla totale inefficienza di questa tecnologia. Nel seguente paragrafo si cercherà di comprendere come sia possibile conciliare la regolamentazione con lo sviluppo di piattaforme aventi come core business la raccolta e la distribuzione dei dati. In particolare, verrà trattato il rapporto tra regolamentazione europea e blockchain.

Nel 2018, la Comunità Europea (CE) ha reso operativo il GDPR (General Data Protection Regulation) al fine di regolamentare la circolazione dei dati sensibili. Il GDPR è una regolamentazione che facilita la circolazione dei dati tra gli Stati Membri della CE ma ne limita la raccolta. Affinché un dato sia trasferito ed immagazzinato deve essere presente il consenso del possessore di dati stessi. Quest'ultimo deve anche indicare quali dati possano essere immagazzinati e per quanto tempo. Inoltre, nel caso in cui i dati dovessero essere trasferiti su un'altra piattaforma, il soggetto deve esserne informato. Tale consenso deve essere esplicito e, soprattutto nelle piattaforme online, si manifesta nella compilazione dei *cookies*. Essi sono dei form da compilare prima dell'accesso al sito in cui si esplicita il consenso sull'utilizzo dei dati personali. In particolare, si indica la motivazione sull'utilizzo dei dati, sull'eventuale trasferimento e sulla possibilità di memorizzazione in strutture dedicate [24].

Il GDPR nasce con l'obiettivo di tutelare gli utenti delle piattaforme che sfruttano i dati forniti dagli stessi. La maggioranza dei social network e delle piattaforme digitali utilizzano questo meccanismo come margine di guadagno. Le principali problematiche derivano dalle loro stesse caratteristiche. I dati, infatti, presentano una natura immateriale che ne facilita la

diffusione, il trasferimento e l'immagazzinamento. In particolare, il GDPR cerca di attenuare tre principali fattori:

- **Data Persistence**: i dati, nel momento della loro generazione, hanno una durata potenzialmente illimitata. Questo è favorito dal loro "immagazzinamento", poiché ha un costo di mantenimento molto basso. Ciò significa che i dati forniti possono essere disponibili per sempre.

- **Data Repurposing**: i dati possono essere riqualificati. Questo comporta l'estrapolazione del dato dal contesto in cui si genera e del suo utilizzato in contesti differenti. I dati forniti possono essere utilizzati per scopi molto diversi da quelli di interesse dell'utente.

- **Data Spillovers**: spesso i dati sono collegati ad altri consentendo di raccogliere informazioni anche non esplicitamente fornite. Ciò significa che soggetti che non hanno fornito i propri dati possono comunque vederli pubblicati poiché collegati ad altri forniti da soggetti che hanno optato per la condivisione.

È evidente come, anche la blockchain, subisca le conseguenze di questa regolamentazione. Il suo network, infatti, funziona soltanto quando è garantita la verifica della consistenza delle transazioni. Tale verifica è possibile grazie alla persistenza ed immutabilità dei dati sul registro, in apparante conflitto con il GDPR. È quindi necessario che i partecipanti alla rete network forniscano l'esplicito consenso all'immagazzinamento ed alla distribuzione dei propri dati, affinché la blockchain rispetti le direttive previste dalla regolamentazione. Spesso questo consenso si assume di default nel momento in cui il nodo partecipa alla rete, ma nella realtà la situazione è molto più complessa [25].

Uno studio della Ontario Tech University ha ipotizzato una prima soluzione per conciliare il GDPR con la tecnologia blockchain [25]. In Figura 3.2 è rappresentato uno schema della struttura ipotizzata. La soluzione proposta prevede l'utilizzo di una permissioned blockchain sviluppata su Hyperledger Fabric. Il risultato è un network gestito da un'autorità terza di cui i nodi hanno fiducia. Oltre allo stesso, è presente un database esterno per la raccolta delle transazioni. Questo è fondamentale poiché la memorizzazione dei dati all'interno della

blockchain ne causerebbe l'immutabilità e la persistenza. La presenza di un database esterno, invece, consente l'eliminazione dei dati ogni qualvolta vi sia una richiesta da parte dei singoli nodi.

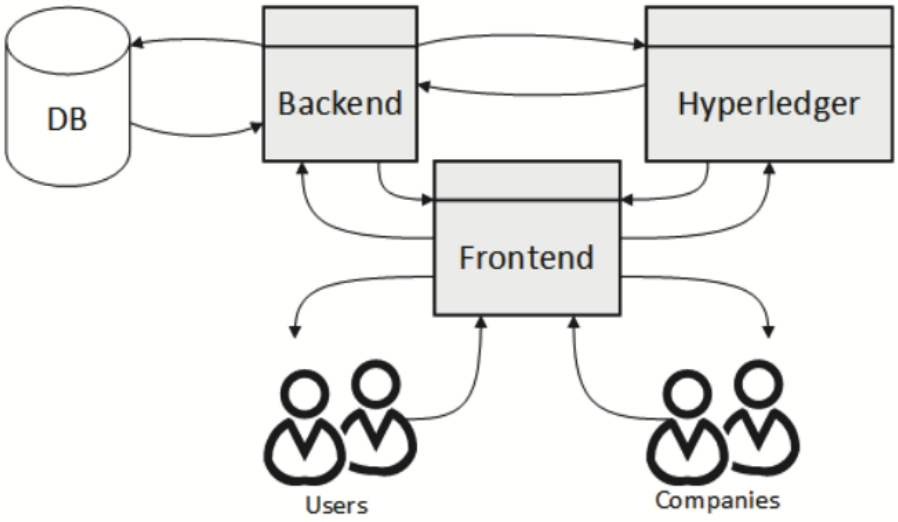

***Figura 3.3**: Struttura Blockchain per il consenso dei dati.*
***Fonte**: Aldred N., Baal L., Broda G., Mahmoud Q.H., Trumble S., Design and Implementation of a Blockchain-based Consent Management System, Oshawa, arXiv, 2019, p. 3*

Come si può vedere in figura, oltre al database e alla rete dei partecipanti, è presente uno stadio intermedio chiamato *REST API Consortium*. Esso è composto da una parte *Frontend*, necessaria per l'inserimento dei dati, ed una parte *Backend*, fondamentale per l'analisi dei dati inseriti dai nodi. L'obiettivo del REST API Consortium è il controllo degli accessi alla blockchain e delle operazioni all'interno di essa. Ogni nodo interagisce con la blockchain attraverso un'architettura REST API che consente di filtrare le richieste dei nodi in base al suo identificativo. La struttura REST è molto utilizzata nelle applicazioni client-server per lo scambio di messaggi. Il protocollo http, ad esempio, utilizzato dalle pagine web come standard comunicativo, si basa proprio su questa tipologia di architettura in cui sono fornite delle risposte in base a specifiche richieste. Ovviamente, soltanto l'autorità superiore conosce la vera identità dei nodi ed il loro scopo nella rete. Ogni accesso da parte di un nodo, prevede l'utilizzo della chiave pubblica (funzione di hash) assegnata dal gestore della rete. Questo consente ad esso di operare all'interno del network mantenendo anonima la sua identità verso gli altri.

Durante il primo accesso alla rete, ogni nodo è invitato ad indicare le modalità con cui i propri dati possano essere trattati e memorizzati. A tale fine, ai partecipanti è proposto un form da compilare per il trattamento dei dati sensibili. Questa prima fase è l'analogo dei cookies nei siti internet, in cui ogni nodo fornisce il proprio "consenso informato" per la partecipazione alla rete. La richiesta del trattamento dei dati personali ha un funzionamento molto simile a quello visto nei social network: più il nodo fornisce consensi, più operazioni può svolgere sulla blockchain e più informazioni sugli altri partecipanti può avere a disposizione. In questo modo si induce il nodo a fornire più dati personali, in cambio di una maggiore libertà di azione sul network.

Il sistema appena descritto consente di rispettare le direttive definite nel GDPR anche nell'utilizzo di una blockchain. L'anonimità dei partecipanti è garantita dall'utilizzo di funzioni di hash e soltanto il gestore dell'infrastruttura è a conoscenza dell'identità dei singoli nodi. Il consenso sull'utilizzo delle informazioni sensibili è richiesto durante il primo accesso ed è l'utente stesso a definire come possano essere utilizzati i propri dati. Infine, la presenza del database, consente la cancellazione dei dati in qualsiasi momento. Tutti i partecipanti hanno il potenziale controllo dei propri dati e possono richiederne la cancellazione. La struttura è in grado così di rispettare questi tre requisiti, garantendo la tutela dei dati personali dei partecipanti senza limitare le potenzialità della blockchain.

3.3 Tipologie di Mapping

Lo studio svolto nei paragrafi precedenti è stato fondamentale per lo sviluppo di tre alternative di *mapping* all'interno della blockchain. Nel contesto informatico, il mapping si riferisce al posizionamento delle informazioni all'interno di un network. Nel caso di studio, il mapping ha l'obiettivo di individuare il miglior posizionamento dei singoli processi all'interno della blockchain. In particolare, si è studiato quali fossero le informazioni dell'iter certificativo da registrare effettivamente nella rete.

Si è già visto come l'iter sia ricco di passaggi che si verificano soltanto al soddisfacimento di determinate condizioni. Nella quasi totalità dei casi le condizioni necessarie sono la compilazione di documenti da parte degli enti in gioco nel processo di certificazione. L'intero processo giunge al termine quando l'organizzazione riceve la certificazione o viene giudicata non idonea e dovrà richiedere un nuovo audit.

Le alternative proposte tengono conto della possibilità di memorizzare le condizioni di passaggio o l'intera documentazione all'interno della blockchain. È importante sottolineare come il caso trattato preveda la presenza di più processi di certificazione in parallelo. La rete è composta da nodi corrispondenti alle organizzazioni che richiedono la certificazione, ed è gestita dall'ente certificatore stesso. Ovviamente, l'ente di certificazione riceve più richieste durante l'anno ed il sistema deve consentire la classificazione dei processi a seconda del richiedente. Per questo tutte le certificazioni presentano un identificativo univoco e, ad ognuno di essi, sono associati una sequenza di codici che identificano i processi interni alla certificazione. Così facendo è sempre possibile risalire a quale organizzazione si faccia riferimento e in quale punto dell'intero iter di certificazione la transazione stia osservando il comportamento. Nei successivi sotto-paragrafi saranno presentate le tre alternative ipotizzate durante la fase di realizzazione del modello. Sarà poi obiettivo dell'ultimo capitolo della tesi individuare la soluzione migliore tra le tre proposte.

3.3.1 Versione A

La prima versione ipotizzata prevede l'introduzione dei soli processi all'interno della blockchain. Le transazioni introdotte dai nodi contengono le informazioni sullo stato di un determinato processo. In particolare, è dichiarato il nodo che ha effettuato l'operazione, l'istante in cui è avvenuta l'approvazione, l'identificativo del processo e della certificazione che si sta analizzando. In Figura 3.3 è rappresentato uno schema di questa prima versione chiamata Versione A.

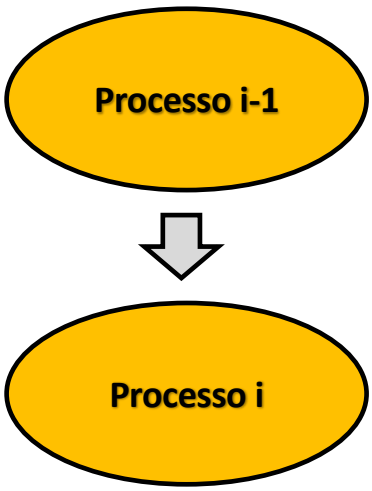

***Figura 3.3**: Schema Mapping Versione A*

I blocchi all'interno della blockchain contengono transazioni che descrivono soltanto il raggiungimento di determinati stati del sistema. Non sono descritte le condizioni che hanno permesso il loro raggiungimento ed a quali documenti si faccia riferimento. Operando su transazioni a contenuto minimo, il network risulta più snello ed efficiente. Dall'altra parte è necessario prevedere l'utilizzo di un database esterno per memorizzare le condizioni ed i documenti, fondamentali per l'analisi della consistenza delle transazioni. Per capire se si tratti o meno dell'alternativa migliore, si dovranno confrontare i vantaggi derivanti da un sistema con elevate performance ed i costi previsti per l'affitto del database.

3.3.2 Versione B

La seconda versione proposta è una soluzione intermedia tra le altre due. Prevede l'introduzione nella blockchain delle condizioni che si sono verificate per il passaggio da un processo ad un altro. In Figura 3.4 è rappresentato lo schema di questa seconda versione chiamata anche Versione B.

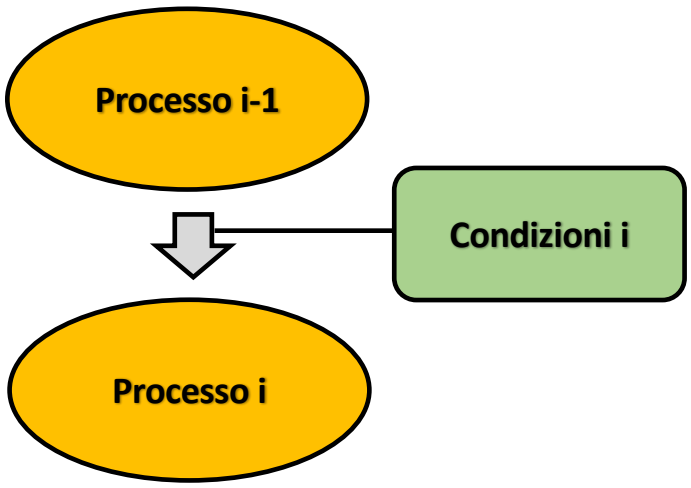

***Figura 3.4**: Schema Mapping Versione B*

La transazione inserita nella rete, oltre a tutte le informazioni già discusse nella versione A, presenta un vettore in cui sono esplicitate le condizioni di passaggio. Dal punto di vista pratico, può essere visto come l'aggiunta di un vettore Booleano rispetto al caso precedente. Esso è un vettore che può assumere soltanto due stati dicotomici (SI-NO, 0-1 etc.) che nel caso in esame sono Vero e Falso. La condizione assume il valore Vero nel momento in cui

essa è verificata e Falso quando è ancora in attesa di una verifica. Per ogni passaggio del processo è noto a priori il numero di condizioni da verificare affinché questo avvenga. La verifica delle condizioni è effettuata analizzando la documentazione necessaria che, anche in questo caso, è memorizzata su un database esterno alla blockchain proprio come nella Versione A.

3.3.3 Versione C

L'ultima versione prevede l'introduzione nella blockchain di tutta la documentazione necessaria per la verifica delle condizioni di passaggio. Ogni transazione contiene i documenti di riferimento in aggiunta a tutte le informazioni già viste nelle versioni precedenti. In Figura 3.5 è schematizzata lo schema di questa terza versione chiamata anche Versione C.

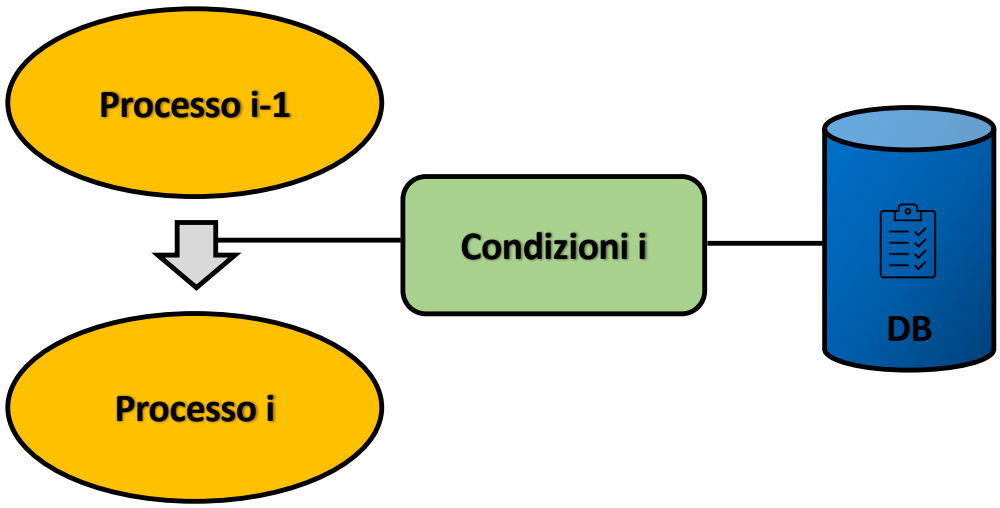

***Figura 3.5**: Schema Mapping Versione C*

Questa terza versione è sicuramente quella con maggiori overhead. Le transazioni presentano delle dimensioni non trascurabili che ne rallentano l'introduzione nella rete ed il loro trasferimento tra i nodi. La blockchain presenta delle performance inferiori per garantire la memorizzazione interna dei documenti ed un risparmio sull'affitto di un database di terze parti. Una soluzione di questo tipo genera un'unica struttura in cui sono presenti tutte le informazioni riguardanti le certificazioni gestite dallo stesso ente, creando quindi un

pacchetto unico già pronto per l'utilizzo. Nel quinto capitolo si cercherà di comprendere i vantaggi derivanti da tale struttura, confrontandola con le due alternative precedenti. Sicuramente è già intuibile come la Versione C sia quella che richieda l'utilizzo di più risorse e di maggiori investimenti nelle infrastrutture.

CAPITOLO 4: Emulatore delle transazioni

4.1 Strumenti di supporto

Nel seguente capitolo si analizzeranno i fattori che hanno consentito la scrittura del software di generazione casuale delle transazioni. L'obiettivo era l'implementazione di un emulatore del processo di certificazione che generasse delle transazioni contenenti le informazioni sul passaggio tra stadi successivi dell'iter stesso. L'utilizzo di tale software è stato fondamentale per testare la permissioned blockchain implementata su Hyperledger Fabric e definirne così le risorse necessarie per il suo sviluppo.

In particolare, si utilizza il termine "emulatore" poiché il software è in grado di emulare il comportamento di un iter certificativo in un ambiente completamente diverso. Le regole seguite, gli input richiesti e gli output ottenuti sono i medesimi di un processo di certificazione, mentre le modalità di esecuzione sono differenti. L'emulatore riesce quindi a riprodurre lo stesso contenuto utilizzando sistemi di sviluppo diversi per i medesimi processi. Nel caso in esame, si emula un iter certificativo senza la presenza di una vera richiesta di certificazione e delle entità viste in precedenza. Per comprendere meglio il concetto di emulatore, si può fare riferimento all'esecuzione di un videogioco arcade (videogioco su macchina a gettoni) in un PC: il gioco è lo stesso su entrambi i dispositivi, ma cambiano le modalità di riproduzione dei processi. Nelle macchine a gettoni, ad esempio, l'interazione avviene grazie a componenti hardware (joystick), mentre il PC ricostruisce i comandi via software. Il processo eseguito è il medesimo (muovere il personaggio nel gioco), nonostante le modalità di esecuzione siano differenti. Lo stesso avviene con l'emulatore delle certificazioni: le transazioni ottenute sono le stesse di un reale processo certificativo nonostante le modalità di ottenimento siano diverse.

Un simulatore, invece, è un software che si comporta in modo simile ad un comune processo, ma è implementato seguendo regole completamente diverse. La difformità non risiede soltanto nell'ambiente di simulazione bensì nella definizione dei processi funzionali. A differenza dell'emulatore, il simulatore copia soltanto il funzionamento di un processo senza mantenerne integre le caratteristiche. Un classico esempio è il simulatore di volo: la

sensazione è la stessa che si prova durante il pilotaggio di un velivolo, ma risulta possibile infrangere delle regole che nel caso reale porterebbero ad errori fatali. Questo poiché, realizzando la simulazione in un ambiente sicuro, è accettabile eseguire delle manovre molto rischiose e non realizzabili nella realtà [27].

Per lo sviluppo del software si è deciso di utilizzare Python come linguaggio di programmazione. La scelta è ricaduta su di esso poiché consente lo sviluppo di programmi complessi senza la necessità di possedere approfondite conoscenze informatiche. Il Python, infatti, utilizza dei costrutti intuitivi e facilmente reperibili su apposite guide. Dovendo interagire con una piattaforma come Hyperledger Fabric che garantisce grande libertà di azione, la scelta di Python come linguaggio è stata quindi immediata.

4.2 Requisiti dell'emulatore

L'analisi svolta nei capitoli precedenti è stato il punto di partenza per lo sviluppo del software. In particolare, si sono esaminati i requisiti necessari per l'emulazione del processo di certificazione. Affinché l'emulatore fosse in grado di riprodurre l'iter certificativo al di fuori dell'ambito delle certificazioni, è stato necessario assumere delle condizioni generali date in input derivanti dall'ambiente in cui opera l'ente di certificazione stesso. In Figura 4.1 è schematizzata la struttura ipotizzata per il funzionamento dell'emulatore.

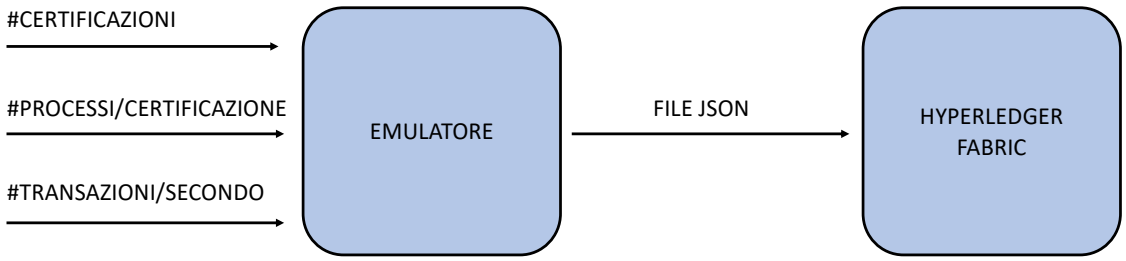

Figura 4.01: *Requisiti dell'emulatore*

Partendo dalla sinistra dell'immagine, si può notare come il software richieda l'introduzione di misure caratteristiche dell'ambito certificativo. Esse variano in base all'ente di certificazione che si sta considerando ed alle sue dimensioni. In particolare, è necessaria la definizione di tre parametri fondamentali:

- **Numero di certificazioni:** certificazioni gestite in parallelo dall'ente certificatore.

- **Numero di processi per certificazione:** processi che caratterizzano le certificazioni amministrate dall'ente. Si ipotizza che tutte abbiano lo stesso numero di processi ed esso sia fornito come valore in input.

- **Numero di transazioni al secondo**: frequenza di generazione delle transazioni da parte dell'emulatore. La definizione del parametro deriva dall'analisi delle capacità della blockchain e dalla velocità di passaggio da un processo all'altro nell'iter certificativo.

L'output dell'emulatore è una sequenza di transazioni che individuano il passaggio da un processo all'altro in una particolare certificazione. Come mostrato nella parte destra della Figura 4.1, tali transazioni sono raccolte su un file JSON inviato poi ad Hyperledger Fabric. Il JSON è un formato adatto all'interscambio di informazioni fra applicazioni client/server, ed è basato su un linguaggio JavaScript. Il grande vantaggio del JSON è la scrittura dei messaggi in un linguaggio comprensibile anche all'uomo, consentendo l'individuazione e la correzione di eventuali errori sopraggiunti durante la comunicazione. Negli ultimi anni il JSON ha riscosso grande successo dato il crescente interesse nello sviluppo di web services [28]. Il formato JSON è molto utilizzato nella fase di comunicazione con le piattaforme come Hyperledger Fabric, per questo è stato scelto come modalità di rappresentazione degli output dell'emulatore.

4.3 Obiettivi dell'emulatore

La scrittura del software è stata possibile definendo sin dal principio gli obiettivi dell'output. Le singole transazioni presenti nel file JSON, infatti, mostrano delle indicazioni puntuali che ne caratterizzano il contenuto e la provenienza. Prima di analizzare tali risultati, è importante sottolineare come il software debba tenere conto delle tre versioni di mapping supposte in precedenza. L'ipotesi sorta durante la fase di progettazione è l'irreversibilità nella scelta del mapping. Se, ad esempio, l'ente di certificazione optasse per la Versione C, tutte le transazioni nel file JSON finale conterrebbero le condizioni ed i documenti necessari per il passaggio al processo successivo. Un'altra considerazione emersa durante la progettazione è la possibilità del software di produrre le transazioni in blocco o in tempo reale. Nel primo

caso, l'emulatore produce direttamente un file JSON contenente tutte le transazioni, riducendo le attese per l'ottenimento dell'output. Questa prima soluzione è stata utilizzata durante la fase di implementazione vera e propria per comprendere la bontà del software sviluppato. La produzione in tempo reale, invece, emula il caso effettivo di introduzione delle transazioni nella blockchain. In questo secondo caso le transazioni sono riportate sul file JSON una per volta, con una frequenza di inserimento prefissata fornita in input.

In Figura 4.2 è rappresentato un esempio di transazione all'interno del file. Il file completo, in realtà, presenta una sequenza di transazioni come quella raffigurata nell'immagine. L'inserimento della transazione nel file JSON corrisponde all'istante in cui essa è introdotta nella blockchain attraverso l'interazione con Hyperledger Fabric. Nel caso di studio, per transazione si intende il passaggio da un processo all'altro di un iter certificativo. Ognuna di esse è caratterizzata da informazioni fondamentali per la loro descrizione: provenienza, istante di approvazione, identificativo e tipologia. In particolare, quest'ultimo aspetto, tiene conto della possibilità di scelta tra le tre alternative di mapping.

***Figura 4.2**: Output dell'emulatore su file JSON*

Dalla figura si può notare come siano presenti diverse indicazioni temporali. La prima, chiamata *istante di ingresso*, definisce l'istante in cui la transazione è inserita nella blockchain. Se si considera il file scritto come blocco unico, il tempo indicato fa riferimento

ad un istante fisso. Nel progetto è stata scelta la data 1/01/2020 alle ore 00:00:00 come istante iniziale. Nel caso di file scritto in tempo reale, invece, l'indicazione temporale è il tempo macchina presente durante l'inserimento. La seconda indicazione temporale è l'*istante di certificazione*, che considera l'istante in cui avviene il passaggio da un processo all'altro nell'iter certificativo. Esso può essere visto come il momento in cui l'ente o l'organizzazione compilano un certo documento.

Nel file sono definiti degli identificatori per il riconoscimento univoco della transazione. È presente un *identificativo transazione* che ha lo scopo di definire in modo sequenziale le transazioni inserite nella blockchain. Ad esempio, se ci si riferisce alla transazione 1, si sta considerando la prima transazione inserita nell'intera rete. L'*identificativo certificazione* e l'*identificativo processo*, invece, hanno la funzione di definire il processo e la certificazione di origine della transazione. Si è ipotizzato un ente certificatore che gestisce un numero di certificazioni individuate univocamente da un valore intero. Ognuna di esse presenta una serie prestabilita di processi interni, individuati anch'essi da un intero sequenziale. Il processo 3 della certificazione 5, ad esempio, indica che la transazione deriva dal completamento del terzo processo del quinto iter certificativo gestito dall'ente.

Nell'output sono presenti anche due stringhe in formato SHA 256. La prima, chiamata *identificativo dichiarante*, definisce il nodo della rete che ha verificato l'autenticità della transazione. La *firma transazione*, invece, è la combinazione di chiave pubblica del mittente e messaggio, utilizzata durante la fase di criptazione asimmetrica. Queste due stringhe sono fondamentali per tracciare il percorso seguito durante l'analisi della consistenza e testimoniare la veridicità della transazione.

I restanti attributi si riferiscono alla tipologia di mapping scelto. La *tipologia di transazione* evidenzia quale delle versioni alternative è stata selezionata dall'ente di certificazione. Nel caso di scelta della Versione B o la Versione C, è presente l'attributo *condizioni* che esplicita le condizioni che si sono verificate per il passaggio allo stadio successivo. Esso è un vettore di condizioni Booleane in cui le celle assumono valore Vero se le condizioni per il passaggio sono rispettate e Falso se non lo sono. Soltanto nel caso della Versione C è presente anche il vettore *documenti,* contenente le informazioni sulla documentazione necessaria per la verifica delle condizioni. Il formato scelto per la raccolta dei documenti è il MIME, formato sviluppato nel contesto della posta elettronica per il trasferimento delle informazioni.

L'obiettivo del MIME è il superamento dei limiti imposti dall'architettura della posta elettronica stessa: messaggi di dimensione limitata, caratteri soltanto in formato ASCII ed impossibilità di inviare allegati e immagini nello stesso messaggio. L'utilizzo di tale formato consente, attraverso una struttura standard, di superare tali limiti a discapito di una maggiore dimensione del messaggio stesso. Con buona approssimazione, si può osservare un aumento della dimensione dei messaggi di un fattore 1,3; ad esempio un file di dimensione 1 MegaByte (MB) è convertito a 1,3 MegaByte.

4.4 L'emulatore

In Appendice I è rappresentato il software completo. Di seguito, invece, è riportata una descrizione delle parti principali, evidenziando le scelte progettuali che hanno condotto a tale soluzione. Non si tratteranno nel dettaglio le strutture base di Python, essendo costrutti standard utilizzati da tutti i programmi sviluppati in questo linguaggio. Inoltre, nell'esempio riportato si analizza la Versione C dato che, includendo il vettore delle condizioni e quello dei documenti, è l'alternativa più completa tra le tre versioni di mapping proposte in precedenza.

In Figura 4.3 è rappresentata la prima parte dell'emulatore, nel quale si definiscono i principali parametri utilizzati.

```
FG = 2
Lmax = 20
NCC = 5000
Tmax = 100000
deltaT = 1/FG
KoT = "C"
t_ID = 0
p_ID = np.zeros(NCC,dtype=int)
data={}
conditions={}
documents={}
type = 'online'
cMax = 10
dMax = 10
char=1000000
```

***Figura 4.3**: Parametri dell'emulatore*

In rosso sono evidenziati i tre parametri visti in Figura 4.1 per il funzionamento dell'emulatore: frequenza di inserimento delle transazioni (*FG*), numero di processi per ogni certificazione (*Lmax*) e numero di certificazioni in parallelo (*NCC*). In particolare, dal parametro FG è ricavato il suo reciproco, chiamato *deltaT*, che indica il tempo trascorso tra l'inserimento di due transazioni successive. Avendo un FG pari a 2 transazioni al secondo, deltaT è pari a 0,5 secondi (1/2). È presente anche il parametro *Tmax* utile per limitare il numero di transazioni inseribili nella blockchain. Esso deriva dall'analisi del tempo massimo per cui la rete può essere occupata dal solo inserimento delle transazioni. Si noti come tale paramatro coincida con il numero di transazioni effettivamente gestite dall'ente (5000x20). Nel caso in esame sono considerate 5000 certificazioni in parallelo, ognuna composta da 20 processi e con frequenza di inserimento pari a 2 transazioni al secondo.

Il parametro *t_ID* è l'identificatore della transazione nella blockchain, mentre *p_ID* è un vettore che individua univocamente i processi delle singole certificazioni. La posizione di ogni cella del vettore corrisponde alla certificazione considerata. Ad esempio, se la cella numero 3 contiene il valore 5, si sta considerando il quinto processo della terza certificazione. In questa prima fase si definiscono anche la variabile *KoT*, identificativa della versione di mapping selezionata, e la variabile *type* che descrive la scelta della generazione in tempo reale o meno dell'output. Nell'esempio si utilizza la Versione C nel caso "online", ossia il file JSON è generato in tempo reale e contiene condizioni e documenti necessari per il passaggio tra processi. A tale scopo sono definiti i parametri *cMax* e *dMax*, che rappresentano rispettivamente il numero massimo di condizioni e di documenti da verificare per il passaggio. In figura si può osservare come entrambi assumono il valore 10, implicando la presenza di massimo dieci condizioni e dieci documenti da analizzare nel passaggio da un processo all'altro.

Nella definizione dei documenti è fondamentale il parametro *char*, che indica il numero di caratteri necessari per rappresentare la documentazione nella blockchain. Durante lo studio dei documenti dell'ente di certificazione si è visto come, utilizzando il formato MIME, essi avessero una grandezza media di 1MB. Questo giustifica la definizione di un char pari a 1000000. Un documento in formato ASCII, infatti, operando su caratteri definiti su 8 bit (1 Byte), necessita della definizione di 1.000.000 di caratteri per ottenere un file finale di 1MegaByte. Infine, sono definiti gli ultimi tre attributi *data*, *conditions* e *documents*, vettori utili per la scrittura dell'output sul file JSON.

L'emulatore vero e proprio comincia con la scelta tra versione online ed offline. In Figura 4.4 sono rappresentate le due alternative, evidenziando in rosso le differenze. Nella versione online, la variabile *T* che rappresenta il tempo di ingresso della transazione nella blockchain, è inizializzata al tempo macchina presente durante l'inserimento. Nel caso offline, invece, T è inizializzata alla mezzanotte del 1° Gennaio 2020. In entrambe le versioni è presente la variabile *dT*, definita in funzione del parametro deltaT, che ha lo scopo di eliminare la linearità assumendo un comportamento Poissoniano. La variabile dT introduce quindi un atteggiamento casuale durante l'inserimento delle transazioni. Rifacendosi all'esempio, un deltaT di 0,5 secondi non implica l'introduzione delle transazioni ad intervalli regolari di 0,5 secondi. La distribuzione di Poisson, infatti, fa sì che l'inserimento sia randomizzato e, ad esempio, si abbia l'introduzione della prima transazione dopo 0,1 secondi e della seconda dopo 0,8 secondi. L'importante è che l'introduzione rispetti una frequenza di inserimento nella blockchain pari a 2 transazioni ogni secondo (ossia quella definita in FG).

```
if type == 'online'
            T=datetime.datetime.now()
            dT =np.random.poisson(deltaT)
            time.sleep(dT)
            T = T + datetime.timedelta (seconds=dT)

if type == 'offline'
             T = datetime.datetime(2020,1,1,00,00,00)
             dT =np.random.poisson(deltaT)
             T = T + datetime.timedelta (seconds=dT)
```

***Figura 4.4**: Differenze tra versione online e offline*

La grande differenza tra le due versioni risiede nella presenza della funzione *time.sleep* nel caso online. Essa consente la generazione del file JSON una transazione per volta, proprio come nel caso reale, con attese durante l'emulazione che scaturiscono dall'arrivo di nuove transazioni. Nel caso offline, invece, il file è generato come un unico blocco, senza la presenza di attese a schermo che emulino il caso reale. Infine, in entrambe le versioni è presente l'aggiornamento della variabile T con il valore di tempo presente dopo l'introduzione della transazione.

Definite le differenze, è presente la parte di software comune alla versione online e offline. In Figura 4.5 è rappresentata la struttura. Questa seconda parte dell'emulatore può essere considerata il corpo centrale del software, in cui si definiscono le parti principali dell'output. La prima riga descrive l'apertura di un file in Python, seguita da un ciclo *while* utile per la scrittura iterativa del file stesso. Il ciclo while è un costrutto utilizzato nella programmazione per rappresentare strutture cicliche che proseguono ad iterare fino a quando non è più verificata una data condizione. Nel caso in esame, la condizione da rispettare è l'inserimento nella blockchain di un numero di transazioni inferiore a Tmax. All'interno del ciclo avviene la scrittura sul file JSON degli attributi delle singole transazioni: tempo di ingresso nella blockchain, tempo in cui è eseguita la certificazione, identificativo della transazione e tipologia di mapping scelto. Inoltre, si utilizza la generazione casuale di stringhe alfanumeriche negli attributi *D_ID* e *Signature*, per definire le funzioni di hash del dichiarante della transazione e la firma digitale presente su di essa.

```
with open ("output.json","w") as outfile:
    while t_ID <= Tmax:
        data['Timestamp']=T
        data['Transaction ID']=t_ID
        data['Kind of Transaction']=KoT
        data['Certification Time']=T
        D_ID = ''.join(random.choices(string.ascii_uppercase + string.digits, k = 32))
        data['Declarant ID']=D_ID
        Signature = ''.join(random.choices(string.ascii_uppercase + string.digits, k = 32))
        data['Signature']=Signature
        found = False
        for i in range(NCC*NCC):
            c_ID = random.randint(0,NCC-1)
            if p_ID[c_ID] < Lmax:
                p_ID[c_ID] += 1
                found=True
                break
```

***Figura 4.5**: Prima fase di scrittura sul File JSON*

In figura, è evidenziata in rosso la struttura tipica del Python per il completamento di un vettore. Si è già discusso come i processi interni ad ogni certificazione siano rappresentati da un vettore in cui ogni cella è un iter certificativo. La struttura evidenziata consente di completare in modo iterativo le singole celle fino al loro completamento, attraverso l'utilizzo di una condizione Booleana assunta Falsa di default. Nell'emulatore tale condizione è

chiamata *found*. In questo caso si utilizza il costrutto iterativo *for* che, a differenza del while, itera un numero prestabilito di volte. Finché la condizione found risulta Falsa, è presente almeno un iter certificativo non ancora ultimato. Nel momento in cui tale condizione diventa Vera, tutti i processi di ogni certificazione risultano completati ed è quindi possibile uscire dal ciclo.

Infine, è presente la sezione dell'emulatore che tiene conto delle tre tipologie di mapping. Anche questa parte è comune sia alla versione online che a quella offline. In figura 4.6 ne è rappresentata la struttura. Se la condizione found risulta Vera, allora è possibile stampare anche il vettore dei processi appena completato. Si può notare come l'identificativo delle transazioni t_ID sia implementato soltanto se found assume il valore Vero. Infatti, se la condizione risultasse Falsa non sarebbe più possibile introdurre transazioni dato che tutti gli iter certificativi risulterebbero ultimati. In questa situazione, il programma prevede l'uscita dal ciclo while e l'arresto dell'emulatore.

```
If found == 'True':
    t_ID += 1
    data['Certification ID']=c_ID
    data['Process ID']=p_ID[c_ID]
     if KoT == 'B' or KoT == 'C':
         Num_conditions = random.randint(1,cMax)
         for i in range (1,Num_conditions):
             conditions['Condition %s'%i]='True'
             data['Conditions']=conditions
             if KoT == 'C':
                 Num_documents = random.randint(1,dMax)
                 for i in range (1,Num_documents):
                     documents['Document %s'%i]=' '.join(random.choices(string.ascii_letters + string.digits, k = char))
                     data['Documents']=documents
     output=json.dump(data,outfile,indent=2,default=str)
else:
    break
```

***Figura 4.6**: Valutazione delle tre alternative di mapping*

La stampa dell'output è eseguita utilizzando l'array data, vettore che si occupa di raccogliere e riportare sul file JSON tutti gli attributi discussi in precedenza. In rosso sono evidenziate le parti del file aggiunte in base alla versione di mapping selezionata. Si può notare come l'identificatore della certificazione e del processo sono stampati qualsiasi sia la versione

scelta, mentre l'array delle condizioni è visualizzato soltanto nelle Versioni B e C. Come già anticipato, esso è un vettore di condizioni Booleane di dimensione cMax in cui le celle assumono valore Vero soltanto quando le condizioni per il passaggio sono rispettate. Ovviamente, affinché il passaggio tra processi possa avvenire, devono essere rispettate un numero sufficiente di condizioni. L'ipotesi fatta nella fase di progettazione è che nell'output siano elencate soltanto quelle che hanno consentito il passaggio, essendo note il numero di condizioni utili per ogni processo. Ad esempio, la presenza di sei condizioni elencate per il passaggio tra il primo ed il secondo processo della quinta certificazione, indica la necessità di verificare sei diversi vincoli per il passaggio.

In Figura 4.6 si può notare come nella Versione C sia presente anche l'array dei documenti, vettore di dimensione dMax contenente in ogni cella il documento in formato MIME. Esso è composto da stringhe alfanumeriche di lunghezza pari a char e, come nel vettore delle condizioni, si ipotizzano noti a priori il numero di documenti necessari per la verifica del passaggio. Ciò implica che ogni iter certificativo presenta un numero massimo di documenti da valutare pari a dMax. Nell'emulatore il numero di condizioni e documenti utili per il passaggio tra processi è definito attraverso l'estrazione casuale di un numero intero tra 1 e cMax o dMax. Questo significa che ad ogni passaggio è possibile richiedere il soddisfacimento di un numero diverso di condizioni anche all'interno della stessa certificazione. Ad esempio, se diversi passaggi della stessa certificazione richiedono la verifica delle condizioni nel documento 3, il documento utilizzato è lo stesso. Questo poiché, come già visto nell'applicazione del modello astratto alla norma ISO 9001, è possibile che processi diversi richiedano firma e controfirma del medesimo documento.

Il software appena descritto fornisce l'output rappresentato in Figura 4.7. In Appendice II è presente uno script completo dell'output. Nell'esempio si considera un'unica transazione nella versione online in cui si utilizza l'alternativa C per il mapping ("Kind of Transaction": "C", definita in precedenza come KoT). In prima approssimazione, si considerano equivalenti l'istante di tempo in cui la transazione è introdotta nella blockchain e l'istante in cui la transazione ottiene il consenso della rete (in figura rappresentati rispettivamente da "Timestamp" e "Certification Time"). Inoltre, per motivi di spazio, char è stato ridotto a 100 caratteri e si considera un ente di certificazione che gestisce 5000 iter certificativi in parallelo.

```
{
 "Timestamp": "2021-02-28 14:31:29.503185",
 "Transaction ID": 0,
 "Kind of Transaction": "C",
 "Certification Time": "2021-02-28 14:31:29.503185",
 "Declarant ID": "3HA15OQ2AHG7II56SOWS7IJH36PXVZTW",
 "Signature": "LSLRZLDAFXFWWEA1OG752FHT6437GTBO",
 "Certification ID": 958,
 "Process ID": "1",
 "Conditions": {
  "Condition 1": "True",
  "Condition 2": "True",
  "Condition 3": "True",
  "Condition 4": "True",
  "Condition 5": "True",
  "Condition 6": "True"
 },
 "Documents": {
  "Document 1":
"Y3FAOCSiU8e155p2L2gAjuUFbtbmknNadB0WHJWaZj0NCUm621u1AL
NbZ8YZhKav7KrMtpOFCc9MgYF3b1cj9TpSsEfKj4kHSuJo"
 }
}
```


***Figura 4.7**: Output emulatore su file JSON*

Il file, essendo in formato JSON, è racchiuso tra due parentesi graffe ed ogni attributo è compreso tra virgolette. Si può notare come la transazione faccia riferimento al processo 1 ("Process ID": "1") della certificazione numero 958 ("Certification ID": 958). Inoltre, essa è la prima introdotta nella blockchain poiché ha come identificativo il valore 0 ("Transaction ID": 0). Sono poi presenti le due stringhe alfanumeriche che rappresentano la funzione SHA 256 del dichiarante della transazione e della firma presente su di essa ("Declarant ID" e "Signature"). Nel vettore conditions si individuano sei diverse condizioni da verificare per il passaggio al secondo processo della certificazione numero 958. Esse sono verificabili attraverso un unico documento riportato nel vettore documents, rappresentato da una stringa alfanumerica di lunghezza char. Nel caso reale, all'interno della transazione è presente il vero documento che ha permesso la verifica delle condizioni, poi memorizzato nella blockchain e sempre disponibile per eventuali controlli.

L'implementazione del software è stata fondamentale per testare la blockchain sviluppata su Hyperledger Fabric. Cambiandone pochi parametri, infatti, è possibile analizzare il diverso comportamento della rete alle tre alternative di mapping. L'interazione dell'emulatore con

la piattaforma ha permesso di estrarre dati e tabelle utili per il confronto di tali alternative. L'obiettivo del prossimo capitolo è approfondire i risultati derivanti dalle tre versioni del software e comprendere quale sia la soluzione migliore in termini di costi, performance e storage.

CAPITOLO 5: Analisi economica delle alternative

5.1 Risultati Sperimentali

Nel seguente capitolo si confronteranno le alternative di mapping ipotizzate in termini tecnico-economici. In particolare, lo studio si concentrerà sul confronto tra Versione A e Versione C. La scelta di escludere la Versione B deriva dalla natura stessa della soluzione: rispetto alla Versione A, infatti, aggiunge soltanto un vettore di condizioni Vero e Falso di dimensione molto piccola rispetto all'intera transazione. Questo implica un'equivalenza tra le prime due alternative che si manifesta in costi del tutto analoghi. La Versione C, invece, presenta anche il vettore dei documenti che comporta un maggiore overhead delle transazioni e costi superiori rispetto alla Versione A, rendendo quindi interessante il confronto.

I dati utilizzati nelle successive analisi derivano da test svolti su una piccola blockchain sviluppata su Hyperledger Fabric, sfruttando l'emulatore discusso nel capitolo precedente. L'emulatore, infatti, ha permesso di analizzare il diverso comportamento della blockchain alle alternative di mapping ipotizzate. Durante questa prima fase di test è consuetudine non investire immediatamente le risorse in infrastrutture fisiche, ma affidarsi a *Virtual Machines (VMs)* presenti sulle piattaforme cloud. Le VMs sono dei computer veri e propri che, a differenza di quelli personali, non sono fisicamente posseduti dal soggetto ma forniscono uno spazio virtuale per lo sviluppo delle applicazioni. Tali risorse sono collocate in opportuni spazi chiamati Data Center, vero cervello di tutti i servizi cloud, composti da migliaia di server in cui sono poste le VMs. La capacità dello spazio virtuale dipende dalle necessità dello sviluppatore ed ha un costo tanto più grande quanta più potenza di calcolo è richiesta. Inoltre, il costo di affitto delle VMs varia a seconda del numero di processori che le compongono, chiamati *Virtual CPU (vCPU)*, e della potenza computazionale selezionata. Il compito dello sviluppatore è l'individuazione all'interno di un determinato Data Center delle VMs che consentono l'implementazione dell'applicazione progettata [29].

L'obiettivo della ricerca è lo studio dei carichi di lavoro presenti durante l'interazione tra le diverse versioni dell'emulatore e la blockchain implementata su Hyperledger Fabric.

L'analisi dei carichi consente l'individuazione delle risorse più appropriate per lo sviluppo della struttura sulle piattaforme cloud messe a disposizione. In altri termini, lo studio confronta i costi delle VMs per l'implementazione delle tre alternative di mapping sul cloud, individuando così la soluzione ottima.

In Tabella 1 sono riportate le condizioni di lavoro ipotizzate. Esse derivano dall'osservazione delle attività di un ente di certificazione di medie dimensioni che gestisce 5.000 certificazioni in parallelo, ognuna composta da 20 processi. Per organizzazioni, come già visto in precedenza, si intendono le imprese che richiedono all'ente una certificazione di qualità.

Numero organizzazioni certificate	5.000
Numero transazioni per organizzazione	20
Numero transazioni all'anno	100.000=5.000x20
Dimensione singolo documento	1MB
Documenti per certificazione	10
Dimensione totale documenti	50GB=5.000x10x1MB

***Tabella 1**: Ipotesi di lavoro*

Si assume lo stesso numero di processi per tutte le certificazioni gestite dal medesimo ente, ricordando come essi corrispondono alle transazioni inserite nella blockchain. I passaggi tra processi sono consentiti grazie all'analisi di opportuna documentazione scambiata tra organizzazione ed ente di certificazione. Nel caso in esame, il numero di documenti utilizzati per ogni certificazione è pari a 10 ed ognuno è caratterizzato da una dimensione di 1MB in formato MIME. In realtà, Hyperledger Fabric non consente l'introduzione di documentazione di dimensione così elevata ma limita la grandezza totale delle transazioni a 200KB. Per questo motivo, nel caso si adoperi la Versione C, i test sono ridotti a documenti di 100KB, implicando delle conseguenze che saranno discusse nel capitolo. Inoltre, riconducendosi a quanto detto per le VMs, eseguire già la fase di test con 100.000 transazioni e 5.000 organizzazioni sarebbe eccessivamente gravoso in termini economici. Per questo, in prima battuta, si è optato per testare una rete di piccole dimensioni composta da solo 4 nodi, nel quale sono introdotte 1.000 transazioni. Le caratteristiche di tale rete sviluppata su Hyperledger Fabric sono riportate in Tabella 2.

Numero peers	3
Numero orderers	1
Totale nodi nella rete	4
Numero CPU	4 vCPU quad-core da 2,4GHz
RAM	8GB
SSD	100GB
Throughput	1tr/s
Transazioni totali	1000

***Tabella 2**: Specifiche tecniche e caratteristiche della rete testata*

La rete testata presenta quattro diversi nodi, tra cui uno di essi è l'ente certificatore. I restanti tre nodi possono essere considerati delle organizzazioni richiedenti la certificazione di qualità all'ente stesso. All'interno della rete sono state introdotte 1.000 transazioni generate casualmente dall'emulatore per comprenderne il funzionamento e l'utilizzo delle risorse. Come appena detto, tale configurazione è molto distante dal caso reale. L'obiettivo però non era la riproduzione fedele del contesto di certificazione, bensì il confronto tra il diverso comportamento della blockchain alle alternative di mapping. Infatti, si può ipotizzare come l'atteggiamento della rete alle alternative rimanga lo stesso anche all'aumento del numero di nodi e delle transazioni inserite. Ovviamente, nel contesto reale, i carichi di lavoro delle VMs ed il traffico generato sono superiori, ma le considerazioni sul comportamento alle alternative rimangono le stesse.

In prima approssimazione, non è stata considerata alcuna differenza tra peers e orderers. In realtà, si è già visto come gli orderers siano dei particolari nodi che si occupano soltanto dell'instradamento del traffico nella rete. Essi presentano carichi di lavoro inferiori rispetto ai peers che, oltre a proporre l'introduzione di nuove transazioni, devono verificare la consistenza e veridicità attraverso il mining. Tale approssimazione si è resa necessaria per disporre di una maggior uniformità nel confronto tra le alternative, non inficiando eccessivamente sui costi finali delle soluzioni.

Le specifiche tecniche delle VMs selezionate tengono conto di due considerazioni: si affittano macchine caratterizzati da unica vCPU e ad ogni nodo è associata una propria VM. Queste assunzioni giustificano la scelta di quattro vCPU nello spazio di sviluppo, corrispondenti proprio a quattro diverse VMs. Le caratteristiche dell'hardware selezionato (vCPU, RAM e SSD) sono standard e consentono lo sviluppo di tutte quelle piattaforme che

non richiedono grandi potenze di calcolo. Inoltre, ogni VM selezionata è soggetta ad una frequenza di inserimento pari a 1 transazione ogni secondo (FG=1tr/s). È importante sottolineare come le specifiche tecniche in Tabella 2 si riferiscono a tutte le VM. Ciò significa che ognuna di esse presenta una vCPU quad-core da 2,4GHz, una RAM da 8GB e un SSD da 100GB.

Testando la blockchain con l'emulatore discusso nel capitolo precedente si ottengono i risultati riportati in Tabella 3.

Dimensione transazione	30KB
Dimensione totale	30MB
Banda richiesta	300KB/s
Utilizzo CPU	7%

(a)

Dimensione transazione	200KB
Dimensione totale	200MB
Banda richiesta	2MB/s
Utilizzo CPU	8%

(b)

Tabella 3*: Caratteristiche transazioni per* ***a)*** *Versione A e* ***b)*** *Versione C*

L'inserimento delle transazioni nella Versione A porta ad un overhead per singola transazione pari a 30KB. Nel caso di Versione C, invece, la dimensione sale fino a 200KB. Tale incremento è giustificato dalla presenza della documentazione necessaria per l'analisi della validità del passaggio in quest'ultima versione. È importante ricordare come nel caso della Versione C le analisi si concentrano su documentazione di dimensione ridotta rispetto a quella effettiva di 1MB. La dimensione totale riportata in tabella tiene conto della grandezza di tutte le 1.000 transazioni generate dall'emulatore. Per quanto riguarda l'utilizzo delle CPU le due versioni si differenziano soltanto per un punto percentuale. Questo è dovuto ad una vCPU con prestazioni decisamente più elevate di quelle necessarie per lo sviluppo della piattaforma e quindi non influenzata dall'incremento delle dimensioni della transazione.

Nel caso della Versione A è necessario prevedere l'affitto di un Database Esterno per tenere traccia dei documenti. La scelta è ricaduta su MongoDB dato che, insieme a CouchDB e LevelDB, è il database più utilizzato da Hyperledger Fabric per la memorizzazione dei dati. In Tabella 4 è descritto il costo mensile di MongoDB considerando il caso reale di documenti da 1MB. Nella configurazione con 4 nodi che gestiscono 1.000 transazioni sono necessari più dei 500MB di storage forniti gratuitamente da MongoDB. Per questo si è selezionato

uno spazio condiviso che garantisce una memorizzazione fino a 2GB. Nelle successive analisi della Versione A si è sempre tenuto conto di questi costi per il database esterno [30].

Memoria garantita	2GB
Costo mensile	7,38€
Costo annuale	88,56€

***Tabella 4**: Costi database esterno MongoDB*

Lo studio svolto nei seguenti paragrafi considera il caso peggiore in cui tutte le transazioni siano inserite nello stesso mese. Tale assunzione impone un'analisi dei costi mensile ed una richiesta di risorse pari a quelle necessarie per gestire tutte le 1.000 transazioni. In particolare, si prevede il confronto di due paradigmi per lo sviluppo cloud della blockchain: rete come *Infrastructure as a Service (IaaS)* o rete come *Platform as a Service (PaaS)*. Nel primo caso si selezionano le risorse necessarie per lo sviluppo autonomo della blockchain. Si studiano quindi le caratteristiche della rete sviluppata su Hyperledger Fabric e si cerca di trasporle nel cloud attraverso la scelta di opportune VMs. Nel caso PaaS, invece, si personalizza una blockchain già sviluppata da un ente terzo a seconda delle proprie esigenze. Il grande vantaggio di quest'ultima soluzione è l'acquisto di un pacchetto completo in cui tutta la struttura di sostegno della rete è già stata definita, con una blockchain già pronta per l'utilizzo. Ovviamente a tali vantaggi corrispondono costi decisamente superiori rispetto al caso IaaS. Si discuteranno nel dettaglio le due alternative valutate su tre cloud provider differenti: Amazon, Aruba e Microsoft Azure.

5.2 Blockchain come IaaS

Nel caso IaaS è necessaria la scelta delle singole risorse utili per lo sviluppo della blockchain sul cloud. Le VMs selezionate per l'implementazione della rete presentano ognuna le caratteristiche riportate in Tabella 2. Inoltre, la banda gestita dalle VMs è nettamente superiore a quella richiesta (ordine dei Gbps). Oltre alla scelta delle risorse è necessario definire le modalità di comunicazione tra le singole VMs ed i rapporti gerarchici presenti tra loro. Si può facilmente intuire come le soluzioni IaaS necessitano di competenze sistematiche non indifferenti per essere implementate correttamente. Questo implica la disponibilità di una risorsa nel team di sviluppo capace di riprodurre tale soluzione, altrimenti non attuabile e da scartare già nella fase di progettazione.

5.2.1 Amazon EC2

Amazon consente lo sviluppo di piattaforme IaaS attraverso lo spazio EC2 di Amazon Web Services (AWS) [31]. AWS è una piattaforma cloud gestita da Amazon che permette lo sviluppo di piattaforma on demand, senza la necessità di acquistare infrastrutture proprie. In particolare, la sezione EC2 si occupa dello sviluppo di tutte quelle applicazioni IaaS in cui è richiesto un elevato livello di personalizzazione. L'implementazione è consentita grazie ad opportuni spazi di sviluppo chiamati *istanze*, ossia porzioni di server dedicate e personalizzabili. Ogni istanza è composta da almeno 2 vCPU e garantisce lo sviluppo di qualsiasi tipo di applicazione. La particolarità di Amazon EC2 è un costo di affitto che non dipende dal numero di istanze selezionate ma dalla quantità di vCPU per cui si è optato. A dimostrazione di ciò, l'affitto di un'unica istanza composta da 4 vCPU ha lo stesso costo di due istanze contenenti ognuna 2 vCPU. Il vantaggio nella scelta di un'istanza singola risiede nello sviluppo della piattaforma in un unico ambiente ottimizzato per lo scambio di informazioni tra i nodi. Dal punto di vista dei soli costi, però, le due soluzioni sono perfettamente interscambiabili.

Il punto di forza di AWS rispetto ai competitors è la definizione dell'utilizzo delle risorse fin dalla selezione delle VMs. Aruba Cloud e Microsoft Azure, invece, consentono di gestire i consumi soltanto dopo un primo periodo di utilizzo del servizio. Nel caso in esame, in cui si è a conoscenza dell'utilizzo effettivo delle risorse (Tabella 3), la scelta di AWS permette quindi un leggero risparmio. È importante ricordare come lo spazio di sviluppo scelto sul cloud faccia riferimento all'implementazione di una blockchain su Hyperledger Fabric, composta da quattro nodi gestiti da un unico proprietario. Nel caso di AWS, i costi sono definiti ipotizzando l'utilizzo di risorse presso il Data Center di Milano ed utilizzando un rapporto di conversione dollaro-euro pari a 0,82 (1$=0,82€). Inoltre, per limitare le spese, si considera una fatturazione online senza cambio mensile delle risorse. Amazon, infatti, predilige una continuità di utilizzo ed applica dei costi aggiuntivi nel caso si decida di optare per il cambio di VMs.

Di seguito si propongono alcune soluzioni alternative per lo sviluppo di tale soluzione sul cloud di AWS. In Tabella 5 è rappresentata una prima analisi dei costi per lo sviluppo della blockchain.

Istanza	Linux c5d.xlarge
Costo istanza	9,73€/m
Costo MongoDB	7,38€/m
Costo totale	**17.11€/m**

(a)

Istanza	Linux c5d.xlarge
Costo istanza	11,04€/m
Costo MongoDB	/
Costo totale	**11,04€ /m**

(b)

Tabella 5*: Costo mensile istanza EC2 per* ***a)*** *Versione A e* ***b)*** *Versione C*

La soluzione sembra in apparente contrasto con quanto ipotizzato: la Versione C risulta più conveniente nonostante presenti delle transazioni con overhead maggiore. In realtà, si devono ricordare i limiti di Hyperledger Fabric citati nei paragrafi precedenti. Infatti, la Versione C si riferisce a transazioni limitate a documenti di dimensioni 100KB che non rappresentano il caso reale in cui la documentazione in formato MIME ha grandezza 1MB. Seguendo tale limitazione, la Versione C è più conveniente soltanto se è realizzabile il ridimensionamento delle transazioni a 200KB. Tutte le analisi fatte da questo momento in poi tengono conto della limitazione. La Versione A, invece, ipotizza il salvataggio di documenti da 1MB sula database esterno MongoDB. Osservando poi la prima riga della tabella, si può notare come la Versione C presenti dei costi superiori per l'affitto della medesima istanza. Questo poiché, nonostante le limitazioni dei documenti, la Versione C ha un utilizzo della CPU comunque superiore rispetto alla Versione A, dovuto al maggiore overhead delle singole transazioni.

L'istanza selezionata presenta delle VMs con sistema operativo Linux. La scelta è ricaduta su questa tipologia essendo la più diffusa ed utilizzata nell'implementazione di applicazioni. L'acronimo "c5d" indica una particolare istanza che facilita lo scambio di informazioni tra le VMs che la compongono, fondamentale per lo sviluppo di una blockchain prestante. La sigla xlarge, invece, evidenzia come la scelta di quattro VMs per un'unica istanza sia un valore relativamente elevato tra le soluzioni possibili.

Una soluzione alternativa proposta da AWS prevede l'utilizzo dell'EBS. Esso è un vero e proprio container comune tra le VMs dell'istanza, in cui è possibile memorizzare in maniera condivisa i dati. In particolare, si è optato per un EBS con snapshot su tutti i dati memorizzati. Lo snapshot implica che all'inserimento di ogni nuova transazione vi sia un'istantanea dello stato del sistema, prevenendo così possibili perdite di dati nel caso di improvvisi malfunzionamenti. Data la presenza dell'EBS nei servizi AWS si è ipotizzata una

soluzione in cui le VMs non presentassero SSD proprie ma un unico spazio di memoria condiviso da 100GB. In Tabella 6 sono riportati i costi mensili della soluzione.

Istanza	Linux c5.xlarge
Costo istanza	8,62€/m
Costo EBS	10,88€/m
Costo MongoDB	7,38€/m
Costo totale	**26,88€/m**

(a)

Istanza	Linux c5.xlarge
Costo istanza	9,77€/m
Costo EBS	10,88€/m
Costo MongoDB	/
Costo totale	**20,65€/m**

(b)

Tabella 6*: Costo mensile istanza EC2 con EBS per* ***a)*** *Versione A e* ***b)*** *Versione C*

Paradossalmente il costo aumenta rispetto alla soluzione precedente. In particolare, decrescono i costi per l'affitto dell'istanza ma aumentano quelli per lo spazio di memoria condiviso. Tale incremento è giustificato dal fatto che l'EBS è gestito direttamente da Amazon, rendendolo quindi altamente scalabile e prestante. Nel caso di SSD per singola VM, invece, è lo sviluppatore dell'applicazione che provvede ad un utilizzo ottimizzato della memoria, oltre alla definizione delle risorse e delle caratteristiche della rete. In questa soluzione si può ipotizzare il registro delle transazioni salvato sull'EBS condiviso tra le risorse. Si può notare come l'assenza della SSD implichi la perdita della lettera "d" al nome dell'istanza. La consonante, infatti, individua proprio la presenza di un hard disk dedicato per ogni VM. In linea generale si ha quindi una riduzione dei costi delle istanze per l'assenza di SSD dedicati ed un aumento dei costi per la memorizzazione ottimizzata delle transazioni. Anche in questa soluzione, come nella precedente, si prevede un backup continuo di tutta la memoria attraverso lo snapshot. Nel caso della Versione A i costi sono maggiorati dall'affitto del database MongoDB per la documentazione da 1MB, rendendo quindi la Versione C più conveniente.

Un' ultima soluzione valutata per AWS è l'affitto di server dedicati, chiamati anche *host*. La differenza con i due casi precedenti è che gli host permettono lo sviluppo di più istanze di clienti diversi. Ciò significa che la scelta di una singola istanza può portare alla condivisione del server su cui è sviluppata con altri utenti del cloud. Optando per l'affitto di interi host, invece, si ha a disposizione un server dedicato per le proprie esigenze, in cui sono già presenti tutte le licenze dei software utili per lo sviluppo delle applicazioni. Quest'ultimo aspetto è molto vantaggioso poiché la gestione sul cloud delle licenze è molto spesso uno dei fattori più critici. In Tabella 7 sono rappresentati i costi per l'affitto di un singolo host.

Istanza	c5d
Costo host	192,48€/m
Costo MongoDB	7,38€/m
Costo totale	**199,96€/m**

(a)

Istanza	c5d
Costo host	218,39€/m
Costo MongoDB	/
Costo totale	**218,39€/m**

(b)

Tabella 7*: Costo mensile host dedicati per* ***a)*** *Versione A e* ***b)*** *Versione C*

Si è optato per l'implementazione sull'host di un'unica istanza composta da 4 vCPU con le stesse caratteristiche discusse in Tabella 5. Da notare, infatti, come sia nuovamente presente la "d" ad indicare la presenza di SSD per ogni VM. Inoltre, nonostante le limitazioni imposte da Hyperledger Fabric, la Versione C ha costi superiori, a causa del maggiore utilizzo dell'host (8% rispetto al 7% della Versione A). Come ci si aspettava i costi sono più che raddoppiati rispetto ai casi precedenti. Tale soluzione, infatti, fornisce delle potenze di calcolo e delle personalizzazioni molto elevate e presumibilmente non necessarie per lo sviluppo di una semplice blockchain di piccole dimensioni. Si può quindi ipotizzare già in questa fase di scartare tale soluzione a vantaggio delle due precedenti.

5.2.2 Aruba Cloud

Aruba è una delle possibili alternative per l'implementazione di piattaforme cloud [32]. In questi ultimi anni Aruba ha raggiunto grandi consensi e si è imposto tra i maggiori cloud provider della regione europea. Inoltre, ad inizio 2021, ha ottenuto dalla CE la certificazione per il rispetto delle linee guida dettate dal GDPR. Tale attestato certifica la conformità dei server alla regolamentazione europea e fornisce un grande vantaggio competitivo, essendo i clienti interessati alla segretezza dei dati e delle soluzioni implementate. Aruba, infatti, si impegna a non diffondere le soluzioni sviluppate sui propri cloud a meno di esplicito consenso degli sviluppatori stessi.

A differenza di AWS, Aruba propone dei pacchetti di VMs già pronti all'uso. La soluzione offerta ha lo scopo di massimizzare le prestazioni delle risorse all'interno degli stessi pacchetti. La personalizzazione è quindi ridotta e non è possibile definire in prima battuta l'utilizzo delle risorse selezionate. Soltanto dopo i primi mesi di utilizzo del servizio, infatti, è possibile gestire i costi in base ai consumi. Questi pacchetti possono essere visti analogamente alle istanze di AWS, essendo degli spazi di sviluppo che presentano delle VMs installate su un determinato server.

La prima soluzione valutata su Aruba Cloud prende il nome di Aruba Extra Large. I costi sono riportati in Tabella 8.

Pacchetto	Extra Large
Costo host	25,00€/m
Costo MongoDB	7,38€/m
Costo totale	**32,38€/m**

(a)

Pacchetto	Extra Large
Costo host	25,00€/m
Costo MongoDB	/
Costo totale	**25,00€/m**

(b)

Tabella 8*: Costo mensile Aruba Extra Large per* ***a)*** *Versione A e* ***b)*** *Versione C*

La soluzione prevede l'affitto di risorse con le solite caratteristiche riportate in Tabella 2. L'unica differenza è la presenza di un SSD da 160GB (non più da 100GB) che non influisce sull'utilizzo di CPU e RAM ipotizzati. Come nel caso di AWS, i minori costi della Versione C derivano dalla limitazione delle transazioni attuata da Hyperledger Fabric. Inoltre, si può notare come i costi totali offerti da Aruba siano di poco superiori alle alternative viste su Amazon. L'incremento di prezzo può essere giustificato dalla mancata definizione a priori dell'utilizzo delle risorse, consentita invece da EC2.

Una seconda soluzione è l'affitto di server dedicati, come già visto per gli host in AWS. La soluzione prende il nome di Arouba Cloud Pro ed i suoi costi sono riportati in Tabella 9.

Server	Linux
Supervisore	Hyper-v
Costo server	93,96€/m
Costo MongoDB	7,38€/m
Costo totale	**100,98€/m**

(a)

Server	Linux
Supervisore	Hyper-v
Costo server	108,00€/m
Costo MongoDB	/
Costo totale	**108,00€/m**

(b)

Tabella 9*: Costo mensile Aruba Cloud Pro per* ***a)*** *Versione A e* ***b)*** *Versione C*

A differenza della versione con unico pacchetto, Aruba Cloud Pro consente una minima personalizzazione del server selezionato. In entrambe le versioni si ipotizza nuovamente una SSD da 100GB e, per i vantaggi già evidenziati nel caso di Amazon, il sistema operativo delle VMs è Linux. Inoltre, Aruba propone anche degli strumenti software per il monitoraggio del server chiamati *Hypervisor* (supervisori). Nel caso in esame si è scelto come supervisore *Hyper-v* che, a fronte di una spesa maggiore, controlla lo stato del server anche durante le fasi di inutilizzo. Tali strumenti software controllano continuamente il

server e segnalano eventuali malfunzionamenti che potrebbero compromettere i dati memorizzati.

I prezzi risultano minori rispetto all'affitto di host su EC2 poiché in AWS sono state selezionate delle VMs adatte alla gestione di traffici elevati. Su Aruba, invece, oltre al sistema operativo e al supervisore non è possibile selezionare nel dettaglio le caratteristiche delle quattro risorse sul server. Esse sono assegnate di default e presentano caratteristiche standard. Anche in questo caso la Versione C presenta un costo superiore rispetto alla Versione A, per gli stessi motivi già visti negli host di AWS. Inoltre, i prezzi sono decisamente maggiori rispetto alla scelta di un pacchetto Aruba Extra Large, rendendo quindi poco efficiente la scelta di tale soluzione.

5.2.3 Microsoft Azure

L'ultimo servizio cloud analizzato è offerto da Microsoft e prende il nome di Azure. Come nel caso di Aruba, anche Azure consente la definizione dell'utilizzo delle risorse soltanto dopo un primo periodo di prova. Inoltre, si ritorna al concetto di istanza visto con AWS. La stima dei costi in questa soluzione prevede l'utilizzo di un Data Center dell'Europa Occidentale, senza la necessità di approssimazioni euro-dollaro poiché i prezzi forniti sono già in valuta europea. I costi per l'affitto di risorse su Azure sono riportati in Tabella 10 [33].

Istanza	Linux D4a v4
Costo istanza	27,45€/m
Costo MongoDB	7,38€/m
Costo totale	**34,83€/m**

(a)

Istanza	c5d
Costo host	27,45€/m
Costo MongoDB	/
Costo totale	**27,45€/m**

(b)

Tabella 10*: Costo mensile istanza Azure per* ***a)*** *Versione A e* ***b)*** *Versione C*

Anche in questo caso si opta per un'istanza con sistema operativo Linux. In particolare, la "Da v4" è un'istanza utilizzata per consentire un grande scambio di dati tra le VMs. Proprio in questo fattore si giustifica il piccolo aumento di prezzo rispetto ad Aruba Extra Large, in cui le risorse fornite avevano carattere generale ed erano applicabili senza grandi differenze a tutte le piattaforme. Il confronto con AWS, invece, deve tenere conto dell'impossibilità di definire a priori i consumi e di conseguenza un prezzo più elevato per Azure. Come sempre, ognuna delle VMs selezionate presenta le specifiche tecniche riportate in Tabella 2. Dal

quadro generale, si può quindi concludere come Azure fornisca dei prezzi perfettamente in linea con i competitors.

5.3 Blockchain come PaaS

Amazon e Microsoft offrono anche un servizio in cui nel cloud è presente la blockchain sviluppata su Hyperledger Fabric. Questo significa che non è compito dello sviluppatore selezionare le VMs necessarie ed implementare la rete, ma è il servizio stesso che si occupa dell'intera gestione. Il compito rimane la personalizzazione della rete in base alle proprie esigenze, indicando il numero di nodi, il traffico generato e lo storage necessario. Questa soluzione è preferibile quando il team di sviluppo non presenta delle competenze approfondite di implementazione sulle piattaforme cloud. Optando per una blockchain già sviluppata dal cloud provider, infatti, si ottiene una soluzione pronta all'utilizzo che gestisce autonomamente il comportamento delle VMs e la comunicazione tra di esse. L'importante è la chiara definizione della struttura della rete e dei carichi di lavoro a cui è soggetta.

In questo caso non sono più presenti le specifiche tecniche utilizzate con IaaS (Tabella 2), ma soltanto le caratteristiche della struttura che si ha intenzione di testare. È importante ricordare come la rete sia testata su quattro nodi, non rappresentando quindi il caso reale con 5.000 organizzazioni gestite in parallelo. L'obiettivo non è l'individuazione dei costi effettivi, ma il confronto tra le diverse alternative per identificare la soluzione migliore. Perciò, tenendo conto del diverso ordine di grandezza, si prendono per buoni i risultati delle seguenti analisi.

Le ipotesi per l'implementazione della blockchain come PaaS sono riportate in Tabella 11.

Numero nodi	4
VM per nodo	1
Dimensione totale transazioni A	30MB
Dimensione totale transazioni C	200MB

***Tabella 11**: Ipotesi sviluppo blockchain come PaaS*

Anche in questa soluzione, per la Versione A, si prevede il posizionamento dei documenti nel database MongoDB esterno alla rete. La Versione C, invece, presenta la solita

limitazione dovuta all'impossibilità di Hyperledger Fabric di introdurre transazioni con dimensione superiore a 200KB. Per questo, anche nello sviluppo come PaaS, si prevede l'approssimazione dei documenti da 1MB a 100KB, comportando per la Versione C dei costi ridotti.

Si può notare come in Tabella 11 non sia riportata alcuna indicazione sulle specifiche tecniche delle VMs. Questo poiché è il cloud provider che seleziona le risorse adeguate in base alle dimensioni della rete ed al traffico in essa generato. L'unica scelta che si effettua è l'attribuzione delle vCPU ad ogni nodo, nonostante non sia possibile definire a priori il loro utilizzo. Nel caso in esame, sia per Amazon che per Azure, si opta per l'assegnazione a tutti i nodi di una sola VM con unica vCPU. Inoltre, si ipotizza la presenza dell'ente certificatore con ulteriori tre nodi che rappresentano le organizzazioni richiedenti la certificazione. L'intera rete gestisce le 1.000 transazioni con cui si è testata la blockchain. Anche in questa soluzione tutti i calcoli riportati si riferiscono a costi mensili, approssimando a 720 le ore presenti in un mese (720h/m=24h/ggx30gg/m). Per le analisi, invece, si utilizzano come dimensione delle transazioni le stesse utilizzate nel caso IaaS, riportate in Tabella 11.

5.3.1 Amazon Blockchain

Nel caso di Amazon si prevede lo sviluppo della blockchain in un Data Center situato nelle vicinanze di Londra, utilizzando un rapporto di conversione dollaro-euro pari a 0,82. I costi ottenuti si riferiscono al caso chiamato "Test Network", ossia al test di una rete in via di sviluppo sui cloud di Amazon. In Tabella 12 è riportato il loro valore unitario [34].

Istanza	bc.t3.small
Costo per partecipante	0,30€/h
Costo per nodo	0,038€/h
Costo storage	0,098€/m per GB
Costo scrittura	0,098€/m per GB

***Tabella 12**: Costi unitari Amazon Blockchain*

La soluzione prevede la scelta di un'istanza con sistema operativo Linux diverso dai casi visti in precedenza per AWS. Il "bc" nel nome, infatti, indica come l'istanza sia dedicata soltanto allo sviluppo delle blockchain e presume l'attribuzione di una VM ad ogni nodo partecipante. In realtà, la "t3.small" crea una vera istanza dedicata per ogni partecipante,

messa poi in comunicazione con tutte le altre. Amazon prevede un costo fisso per i partecipanti della rete e per i nodi ad essi dedicati. Il costo di partecipazione è decisamente più alto rispetto a quello del nodo poiché la presenza di nuovi partecipanti implica la ridefinizione dell'intero organigramma. Inoltre, è presente un costo fisso per ogni GigaByte scritto e memorizzato nella rete stessa.

L'ipotesi di lavoro è la presenza di un ente di certificazione e tre diverse organizzazioni che richiedono l'attestato. Questo comporta la presenza di quattro partecipanti ad ognuno dei quali è assegnato un singolo nodo. Considerando le 720h/m ipotizzate e moltiplicando i costi unitari per i valori riportati in Tabella 11, si ottengono i costi mensili presenti in Tabella 13.

Costo partecipanti	846,72€/m
Costo nodi	89,28€/m
Costo storage	0,012€/m
Costo scrittura	0,003€/m
Costo MongoDB	7,38€/m
Costo totale	**943,40€/m**

(a)

Costo partecipanti	846,72€/m
Costo nodi	89,28€/m
Costo storage	0,078€/m
Costo scrittura	0,020€/m
Costo MongoDB	/
Costo totale	**936,10€/m**

(b)

Tabella 13*: Costo mensile Amazon Blockchain per* ***a)*** *Versione A e* ***b)*** *Versione C*

Il costo finale dei partecipanti si ottiene moltiplicando il costo unitario in Tabella 12 per le 720h/m previste ed il numero di partecipanti alla rete. Stesso ragionamento vale anche per il costo dei nodi. Nonostante storage e scrittura presentino lo stesso costo unitario, il prezzo finale è differente. Per lo storage, infatti, si considera la memorizzazione del registro presso ogni nodo ed è quindi necessario moltiplicare il costo mensile per i quattro nodi previsti. Nella scrittura, invece, si deve soltanto tenere traccia della dimensione totale dei dati introdotti e non dei nodi che effettuano l'inserimento. Questo giustifica un prezzo finale della scrittura pari a un quarto di quello dello storage (manca appunto il prodotto con i quattro nodi della rete). Anche in questo caso la Versione C presenta dei costi minori a causa della limitazione dei documenti.

Si può notare come i costi per una soluzione PaaS siano di molto superiori al caso IaaS. Questo è anche giustificato dal fatto che tali costi non tengano conto dell'impiego delle risorse, definibile soltanto dopo un primo periodo di utilizzo. Ipotizzando un utilizzo del 7%

per la Versione A ed un 8% per la Versione C, infatti, i costi mensili sarebbero certamente inferiori e direttamente confrontabili con quelli della soluzione IaaS.

5.3.2 Azure Blockchain

Nel caso di Microsoft Azure si opta per un Data Center posizionato nell'Europa Occidentale, in cui i costi sono già forniti in euro e non è quindi necessaria la conversione vista in precedenza. In Tabella 14 sono riportati i costi unitari per questa soluzione [35].

Istanza	Basic
Costo per nodo	0,340€/h
Costo storage	0,053€/m per GB
Costo scrittura	0,0002€/tr

***Tabella 14**: Costi unitari Azure Blockchain*

L'istanza Basic è utilizzata da Azure per testare il funzionamento di una rete in via di sviluppo. Si può notare come Microsoft, a differenza di Amazon, non faccia pagare un costo fisso ai partecipanti della rete, ma assegni dei costi per singolo nodo decisamente più elevati rispetto a quelli riportati in Tabella 12. Infatti, il prezzo per nodo in Azure è circa uguale alla somma del costo di partecipazione (0,30€/h) ed il costo per nodo (0,038€/h) visti in Amazon. Inoltre, l'approccio utilizzato da Microsoft è quello di far pagare una quota fissa per ogni transazione scritta indistintamente dalla sua dimensione. Questo giustifica il costo definito sul numero di transazioni e non sulle dimensioni totali. Per quanto riguarda la gestione dello storage, invece, Azure prevede dei costi minori rispetto ad Amazon. Considerando le 720h/m ed i quattro nodi della rete si ottengono i costi in Tabella 15.

Costo nodi	979,20€/m
Costo storage	0,006€/m
Costo scrittura	0,20€/m
Costo MongoDB	7,38€/m
Costo totale	**986,79€/m**

(a)

Costo nodi	979,20€/m
Costo storage	0,043€/m
Costo scrittura	0,20€/m
Costo MongoDB	/
Costo totale	**979,44€/m**

(b)

***Tabella 15**: Costo mensile Azure Blockchain per **a)** Versione A e **b)** Versione C*

Si può notare come i costi siano superiori rispetto alla soluzione proposta da Amazon. Questo trova spiegazione nel diverso approccio alla scrittura sulla blockchain: per Azure,

infatti, non è importante la dimensione totale delle transazioni ma quante ne sono state introdotte. Non conta quindi se la dimensione totale è di 30MB come nella Versione A o 200MB come nella Versione C. Infatti, essendo 1.000 le transazioni in entrambe le versioni, i costi di scrittura sono gli stessi e non vi è la taratura del traffico vista invece nel caso di Amazon. Inoltre, Microsoft offre un costo di partecipazione alla rete superiore. Per quanto riguarda il costo dei nodi e dello storage, invece, i ragionamenti sono analoghi a quelli visti per Amazon. Anche in questo caso, tenendo conto dell'impossibilità in fase di test di definire l'utilizzo delle risorse, la soluzione presenta dei costi decisamente superiori al caso IaaS.

5.4 Analisi dei risultati

Le analisi svolte nei paragrafi precedenti consentono di trarre delle importanti conclusioni. In primo luogo, si può affermare come la scelta del mapping dipenda da fattori esterni non controllabili direttamente da chi implementa la rete. Infatti, la limitazione imposta da Hyperledger Fabric sulla dimensione massima delle transazioni, comporta la scelta della Versione A nel caso in cui i documenti per il controllo delle condizioni superino il limite di 200KB. In questo caso è quindi necessario l'affitto di un database esterno con gli annessi costi mensili. Diversamente, se vi è la possibilità comprimere la dimensione dei documenti e rispettare i vincoli di Hyperledger Fabric è preferibile l'utilizzo della Versione C. In tale alternativa di mapping, infatti, si ha un unico spazio contenente tutte le informazioni riguardanti i processi di certificazione del medesimo ente. In questo modo si ha un'ottimizzazione della rete a scapito di una maggiore occupazione delle risorse e di traffico generato che conduce, come visto nei paragrafi precedenti, a costi minori rispetto alla Versione A.

Un'altra importante considerazione riguarda le modalità di sviluppo. La soluzione IaaS richiede competenze tecniche per la scelta delle risorse, consentendo però dei risparmi non indifferenti. L'alternativa PaaS, invece, permette l'acquisto di un pacchetto completo già pronto all'utilizzo e monitorato dal cloud provider stesso, a scapito di costi decisamente più elevati. La scelta tra le due alternative dipende quindi dalle competenze del team di sviluppo e dai suoi obiettivi. Se l'organizzazione presenta le competenze ed è stata ingaggiata per la gestione dello sviluppo cloud, la scelta IaaS risulta quella più opportuna. Contrariamente, se l'organizzazione si occupa di tutta la trasposizione dei processi certificativi sulla blockchain e non presenta le competenze di sviluppo cloud, la soluzione PaaS è certamente la migliore.

Inoltre, all'interno della stessa soluzione è compito dell'organizzazione scegliere il cloud provider più opportuno. Nell'IaaS, ad esempio, si è visto come i costi proposti da Amazon, Aruba e Azure siano perfettamente in linea tra di loro. L'unica differenza risiede vantaggio di AWS di definire a priori l'utilizzo delle risorse affittate, colmato però da alcuni mesi di utilizzo di Aruba e Azure. La scelta dell'alternativa più idonea ricade quindi in mano al committente dell'intera struttura. Ugual discorso vi è nella scelta tra Amazon e Azure per lo sviluppo della blockchain come PaaS.

Nel caso di studio si cerca di applicare la blockchain al contesto delle certificazioni e si vuole individuare la soluzione ottima. A tale scopo, per prima cosa, si deve verificare la possibilità di ridurre la dimensione dei documenti utili per l'analisi delle condizioni di passaggio. Se la compressione è possibile, la Versione C risulta praticabile e quindi preferibile, altrimenti l'unico mapping attuabile è quelle previsto nella Versione A. Tutto risiede quindi nel valutare la fattibilità della riduzione dei documenti da 1MB ad almeno 100KB per rispettare i vincoli di Hyperledger Fabric. Per la scelta dello sviluppo sul cloud, invece, dipende dalle competenze e dagli obiettivi del team commissionato per lo sviluppo della blockchain. Se il compito del team fosse la gestione dell'intera digitalizzazione del processo di certificazione, non avrebbe senso perdere troppo tempo nell'implementazione delle risorse cloud, preferendo quindi la soluzione PaaS. Viceversa, se l'organizzazione è maggiormente impegnata nello sviluppo cloud della blockchain, è sicuramente consigliata l'alternativa IaaS che permette anche un considerevole risparmio. Tenendo conto di queste considerazioni, sarà comunque il committente (ente certificatore) a decidere quale sia la soluzione migliore in base a costi ed obiettivi.

CONCLUSIONI

Il lavoro di tesi ha permesso di definire alcuni aspetti fondamentali per lo sviluppo di una blockchain nel contesto delle certificazioni. Uno dei maggiori contributi è stato la descrizione del modello astratto generale valido per tutte le certificazioni. Tale modello, infatti, ha consentito l'individuazione dei processi costitutivi degli iter certificativi, divenuti poi transazioni inserite all'interno della blockchain. L'analisi dei processi ha quindi condotto alla valutazione di tre possibili mapping per il posizionamento nella rete della documentazione utile per l'autentificazione delle certificazioni. Al fine di selezionare l'alternativa migliore, si è implementato su Python un emulatore dei processi di certificazione per testare il diverso comportamento della rete nelle tre versioni. In particolare, l'emulatore è stato messo in comunicazione con una piccola blockchain sviluppata su Hyperledger Fabric, caratterizzata da una struttura elementare di soli quattro nodi. La soluzione ottima è stata quindi identificata confrontando i costi di diversi cloud provider per l'implementazione della blockchain come Infrastructure as a Service (IaaS) o Platform as a Service (PaaS).

Lo studio ha condotto all'individuazione della terza versione di mapping (chiamata Versione C) come alternativa migliore. Tale conclusione è valida soltanto nel caso in cui fosse possibile ridurre la dimensione della documentazione in ingresso, così da rispettare i limiti imposti da Hyperledger Fabric. Diversamente, l'unica soluzione praticabile sarebbe la prima versione di mapping (chiamata Versione A) in cui la documentazione è memorizzata esternamente alla rete. Per il confronto tra IaaS e PaaS, invece, la scelta ottimale dipende dalle capacità del team di sviluppo e dalla disponibilità economica del committente. La soluzione IaaS, infatti, richiede maggiori conoscenze tecniche delle risorse cloud ma minore dispendio di tempo e denaro. Contrariamente, la soluzione PaaS offre una soluzione già pronta a fronte di costi decisamente superiori. Questa scelta, a differenza del mapping, è più legata ad una questione progettuale che soltanto il committente può definire.

L'elaborato evidenzia anche i vantaggi derivanti dall'utilizzo della blockchain, in linea con la letteratura esistente. La tecnologia, infatti, consente l'utilizzo di un'architettura distribuita in tutte le reti in cui non è presente la fiducia reciproca tra i partecipanti (caso Bitcoin).

Nonostante ciò, la soluzione proposta dalla blockchain è molto simile a quella di un database distribuito. Anzi, in circostanze simili le performance garantite da quest'ultimo sono decisamente superiori rispetto a quelle offerte dalla blockchain. Questo per sottolineare come, in qualsiasi applicazione, sia fondamentale la comprensione del contesto. Infatti, l'applicazione di una blockchain verrebbe meno in un ambiente in cui sia presente la mutua fiducia, poiché con l'utilizzo di database distribuito si otterrebbero prestazioni più elevate. Viceversa, l'impiego di un database distribuito in un ambiente ostico come quello di Bitcoin non sarebbe attuabile, proprio per la mancanza di reciproca fiducia e di un'autorità superiore che gestisce il rapporto tra partecipanti. La blockchain è quindi una tecnologia molto potente ed applicabile nei settori più disparati, l'importante è contestualizzare sempre la scelta dell'una o dell'altra tecnologia.

Nonostante i rilevanti risultati ottenuti, è importante tenere presente come questa ricerca si sia concentrata maggiormente sulla definizione del modello astratto e dell'emulatore delle transazioni. Le analisi economiche finali, infatti, sono state eseguite utilizzando una blockchain primordiale, caratterizzata da una rete molto distante dal caso reale. Un'eventuale ricerca futura dovrebbe concentrarsi maggiormente sulla definizione di una rete più simile a quella gestita dall'ente certificatore, così da ottenere delle stime dei costi più accurate.

BIBLIOGRAFIA E SITOGRAFIA

1)Alepis E., Casino F., Patsakis C., Politou E. Blockchain Mutability: Challenges and Proposed Solutions. IEEE [Internet]. 2019 [pubblicato il 25 Ottobre 2019; consultato: Gennaio 2021]; 1(1): 1-13. Disponibile all'indirizzo: https://arxiv.org/pdf/1907.07099.pdf

2) Akutsu A., Fujimura S., Kishigami J., Miyazaki Y., Nakadaira A., Watanabe H. Blockchain contract: Securing a blockchain applied to smart contracts. Articolo presentato in: IEEE International Conference on Consumer Electronics, Las Vegas, USA, Giugno 2016.

3) Tang L, Törngren M., Wang L. A Permissioned Blockchain Based Feature Management System for Assembly Devices. IEEE [Internet]. 2020 [pubblicato il 19 Ottobre 2020; consultato: febbraio 2021]; 8(2): 183378-183390. Disponibile all'indirizzo: https://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=9212381

4) Chen X., Dai H-N., Wang H., Xie S., Zheng Z. Blockchain challenges and opportunities: a survey. Int. J. Web and Grid Services [Internet]. 2018 [pubblicato il 28 Ottobre 2018; consultato: Gennaio 2021]; 14(4): 354-355. Disponibile all'indirizzo: https://www.researchgate.net/profile/Hong-Ning_Dai/publication/328271018_Blockchain_challenges_and_opportunities_a_survey/links/5bd2706f92851c6b278f31eb/Blockchain-challenges-and-opportunities-a-survey.pdf

5) Liu M., Wu K., Xu J.J. How Will Blockchain Technology Impact Auditing and Accounting: Permissionless versus Permissioned Blockchain. AAA [Internet]. 2019 [pubblicato nell'Agosto 2019; consultato: Gennaio 2021]; 13(2): 21-22. Disponibile all'indirizzo: https://www.researchgate.net/profile/Kean_Wu/publication/335472340_How_Will_Blockchain_Technology_Impact_Auditing_and_Accounting_Permissionless_Vs_Permissioned_Blockchain/links/5e270a3e299bf15216707ef4/How-Will-Blockchain-Technology-Impact-Auditing-and-Accounting-Permissionless-Vs-Permissioned-Blockchain.pdf

6) Dimitrov D.V. Blockchain Applications for Health Care Data Management. Healthcare Informatics Research [Internet]. 2019 [pubblicato nel Gennaio 2019; consultato: Gennaio 2021]; 25(1): 51-56. Disponibile all'indirizzo: https://synapse.koreamed.org/upload/SynapseData/PDFData/1088hir/hir-25-51.pdf

7) Crosby M., Kalyanaraman V., Nachiappan M., Pattanayac P., Verma S. Blockchain Technology: Beyond Bitcoin. AIR [Internet]. 2016 [pubblicato nel Giugno 2016; consultato: Gennaio 2021]; 1(2): 5-20. Disponibile all'indirizzo: https://j2-capital.com/wp-content/uploads/2017/11/AIR-2016-Blockchain.pdf

8) Velde F. R. Bitcoin: A primer. Chicago Fed Letter [Internet]. 2013 [pubblicato nel Dicembre 2013; consultato: Gennaio 2021]; 1(317): 1-4. Disponibile all'indirizzo: http://blog.philippe-poisse.eu/public/monnaie_locale/bitcoin/cfldecember2013_317.pdf

9) Soldavini P. Il bitcoin non si ferma più: dopo Tesla, sarà Apple a scommetterci?. Il Sole 24ore [Internet].2021[pubblicato il 9 Febbraio 2021; consultato: Febbraio 2021]. Disponibile all'indirizzo: https://www.ilsole24ore.com/art/il-bitcoin-non-si-ferma-piu-tesla-sara-apple-scommetterci-ADCLfuIB

10) Chen G., Dinh T.T.A., Lin Q., Loghin D., Ooi B.C., Ruan P., Zhang M. Blockchain vs Distributed Database: Dichotomy and Fusion. arXiv.org [Internet]. 2021 [pubblicato il 18 Gennaio 2021; consultato: Febbraio 2021]; 1910.01310: 3-18. Disponibile all'indirizzo: https://arxiv.org/pdf/1910.01310.pdf

11) Jourjon G., Nguyen T.S.L., Potop-Butucaru M., Thai K.L. Impact of network delays on Hyperledger Fabric. arXiv.org [Internet]. 2019 [pubblicato il 21 Marzo 2019; consultato: Febbario 2021]; 1903.08856: 1-3. Disponibile all'indirizzo: https://arxiv.org/pdf/1903.08856.pdf

12) Androulaki E., Barger A., Bortnikov V., Cachin C., Chrstdis K., De Caro A., Enyeart D., Ferris C., Laventman G., Manevich Y., Muralidharan S., Murthy C., Nguyen B., Sethi M., Singh G., Smith K., Sorniotti A., Stathakopoulou C., Vukolic M., Weed Cocco S., Yellick J. Hyperledger fabric: a distributed operating system for permissioned blockchain. Articolo presentato in: Thirteenth EuroSys Conference, Porto, Portogallo, Aprile 2018.

13) Agrawal D., Dittrich J., Schuhknecht F.M., Sharma A. How to Databasify a Blockchain: the Case of Hyperledger Fabric. arXiv.org [Internet]. 2018 [pubblicato il 31 Ottobre 2018; consultato: Febbraio 2021]; 1810.13177: 1-8. Disponibile all'indirizzo: https://arxiv.org/pdf/1810.13177.pdf

14) Barger A., Baset S., Dillemberg D., Manevich Y., Novotny P., Zhang Q. LedgerGuard: Improving Blockchain Ledger Dependability. arXiv.org [Internet]. 2018 [pubblicato il 3 Maggio 2018; consultato: Febbraio 2021]; 1805.01081: 1-8. Disponibile all'indirizzo: https://arxiv.org/pdf/1805.01081.pdf

15) Franceschini F., Galetto M., Maisano D.A., Mastrogiacomo L. Ingegneria della Qualità. 4. Torino: C.L.U.T.; 2019.

16) UNI EN ISO 9001:2015, Sistemi di gestione per la qualità. Requisiti, UNI, Milano.

17) https://www.accredia.it/accreditamento/ (collegamento: Febbraio 2021), Sito Ufficiale Accredia

18) Conti T., De Risi P. Manuale della qualità. 1. Milano: il Sole 24 Ore Libri; 2001.

19) Chakrabarti A., Chaudhuri A.K. Blockchain and its Scope in Retail. IRJET [Internet]. 2017 [pubblicato nel Luglio 2017; consultato: Febbraio 2019]; 4(7): 3053-3056. Disponibile all'indirizzo: https://www.irjet.net/archives/V4/i7/IRJET-V4I7616.pdf

20) Chernikova A., Lebedeva T., Livintsova M., Yakovlev A. Improving the quality management system of goods and services based on the Blockchain concept implementation and quality assessment in the digital economy. Articolo presentato in: E3S Web of Conferences, San Pietroburgo, Russia, 2019.

21) Nigischer C., Stjepandic J. Copyright Protection in Additive Manufacturin with Blockchain Approach. Articolo presentato in: 24th ISPE Inc. International Conference on Transdisciplinary Engineering, Singapore, Singapore, Luglio 2017.

22) Mo-Ching Yeung S. Corporate Social Reponsibility and Quality Management System in the Context of Blockchain. Corporate Ownership & Control [Internet].2018 [pubblicato nell'estate 2018; consultato: Febbraio 2021]; 15(4): 231-242. Disponibile all'indirizzo: https://virtusinterpress.org/IMG/pdf/cocv15i4c1p10.pdf

23) Fiorenzo F. Dai prodotti ai servizi. 1. Torino: UTET; 2001.

24) Cataleta A., Longo A., Natale R. GDPR, tutto ciò che c'è da sapere per essere in regola. AgendaDigitale [Internet]. 2020 [pubblicato il 17 Luglio 2020; consultato: Febbraio 2021]. Disponibile all'indirizzo: https://www.agendadigitale.eu/cittadinanza-digitale/gdpr-tutto-cio-che-ce-da-sapere-per-essere-preparati/

25) Finck M. Blockchain and the General Data Protection Regulation. EPRS [Internet]. 2019 [pubblicato nel Luglio 2019; consultato: Febbraio 2021]; 634.445: 8-13. Disponibile all'indirizzo: https://www.europarl.europa.eu/RegData/etudes/STUD/2019/634445/EPRS_STU(2019)634445_EN.pdf

26) Aldred N., Baal L., Broda G., Mahmoud Q.H., Trumble S. Design and Implementation of a Blockchain-based Consent Management System. arXiv [Internet]. 2019 [pubblicato il 20 Dicembre 2019; consultato: Febbraio 2021]; 1912.09662: 1-6. Disponibile all'indirizzo: https://arxiv.org/pdf/1912.09882.pdf

27) https://qastack.it/software/134746/whats-the-difference-between-simulation-and-emulation (collegamento: Febbraio 2021)

28) https://www.json.org/json-it.html (collegamento: Febbraio 2021), Sito Ufficiale JSON

29) Greenberg A., Hamilton J., Maltz D.A., Patel P. The Cost of a Cloud: Research Problems in Data Center Networks. ACM SIGCOMM Computer Communication Review Internet]. 2008 [pubblicato nel Dicembre 2008; consultato: Marzo 2021]; 39(1): 68-73. Disponibile all'indirizzo: https://dl.acm.org/doi/pdf/10.1145/1496091.1496103

30) https://www.mongodb.com/pricing (collegamento: Marzo 2021), Sito Ufficiale MongoDB

31) https://calculator.s3.amazonaws.com/index.html (collegamento: Marzo 2021), Sito Ufficiale Amazon Web Services (AWS)

32) https://www.arubacloud.com (collegamento: Marzo 2021), Sito Ufficiale Aruba Cloud Italia

33) https://azure.microsoft.com/it-it/pricing/details/virtual-machines/linux/ (collegamento: Marzo 2021), Sito Ufficiale Microsoft Azure Italia

34) https://aws.amazon.com/it/managed-blockchain/pricing/hyperledger/ (collegamento: Marzo 2021), Sito Ufficiale Amazon Web Services (AWS)

35) https://azure.microsoft.com/en-us/pricing/details/blockchain-service/ (collegamento: Marzo 2021), Sito Ufficiale Microsoft Azure

ALLEGATO 1: Emulatore Completo

```
import datetime
import random
import time
import numpy as np
import string
import json

FG = 2
Lmax = 1
NCC = 1000
Tmax = 10000
deltaT = 1/FG
KoT = "C"
t_ID = 0
p_ID = np.zeros(NCC,dtype=int)
data = {}
conditions = {}
documents = {}
type = 'online'
cMax = 10
dMax = 10
char = 1000000

if type == 'online':
    T = datetime.datetime.now()
    with open ("output.json","w") as outfile:
        while t_ID <= Tmax:
            data ['Timestamp'] = T
            data ['Transaction ID'] = t_ID
            data ['Kind of Transaction'] = KoT
```

```
data ['Certification Time'] = T
dT = np.random.poisson(deltaT)
time.sleep(dT)
T = T + datetime.timedelta (seconds=dT)
D_ID = ' '. join (random.choices(string.ascii_uppercase + string.digits, k = 32))
data ['Declarant ID'] = D_ID
Signature = ' '. join (random.choices(string.ascii_uppercase + string.digits, k = 32))
data ['Signature'] = Signature
found = False
for i in range (NCC*NCC):
    c_ID = random.randint(0,NCC-1)
    if p_ID[c_ID] < Lmax:
        p_ID[c_ID] += 1
        found=True
        break
if found == True:
    t_ID += 1
    data ['Certification ID'] = c_ID
    data ['Process ID'] = p_ID[c_ID]
    if KoT == 'B' or KoT == 'C':
        Num_conditions = random.randint(1,cMax)
        for i in range (1, Num_conditions):
            conditions ['Condition %s' %i] = 'True'
        data ['Conditions'] = conditions
        if KoT == 'C':
            Num_documents = random.randint(1,dMax)
            for i in range (1, Num_documents):
                    documents ['Document %s' %i] = ' '.
                    join(random.choices(string.ascii_letters    + string.digits, k = char))
                    data ['Documents'] = documents
    output=json.dump(data,outfile,indent=2,default=str)
    outfile.write('\n\n\n')
else:
    break
```

```
if type == 'offline':
  T = datetime.datetime(2020,1,1,00,00,00)
   with open ("output.json","w") as outfile:
      while t_ID <= Tmax:
         data ['Timestamp'] = T
         data ['Transaction ID'] = t_ID
         data ['Kind of Transaction'] = KoT
         data ['Certification Time'] = T
         dT = np.random.poisson(deltaT)
         T = T + datetime.timedelta (seconds=dT)
         D_ID = ' '. join (random.choices(string.ascii_uppercase + string.digits, k = 32))
         data ['Declarant ID'] = D_ID
         Signature = ' '. join (random.choices(string.ascii_uppercase + string.digits, k = 32))
         data ['Signature'] = Signature
         found = False
         for i in range (NCC*NCC):
            c_ID = random.randint(0,NCC-1)
            if p_ID[c_ID] < Lmax:
               p_ID[c_ID] += 1
               found=True
               break
         if found == True:
            t_ID += 1
            data ['Certification ID'] = c_ID
            data ['Process ID'] = p_ID[c_ID]
            if KoT == 'B' or KoT == 'C':
               Num_conditions = random.randint(1,cMax)
               for i in range (1, Num_conditions):
                  conditions ['Condition %s' %i] = 'True'
               data ['Conditions'] = conditions
               if KoT == 'C':
                  Num_documents = random.randint(1,dMax)
                  for i in range (1, Num_documents):
```

```
                documents ['Document %s' %i] = ' '. join
                (random.choices(string.ascii_letters + string.digits, k = char))
                data ['Documents'] = documents
    output=json.dump(data,outfile,indent=2,default=str)
    outfile.write('\n\n\n')
else:
    break
```

ALLEGATO 2: Output emulatore


```
{
  "Timestamp": "2021-03-01 14:46:23.016736",
  "Transaction ID": 0,
  "Kind of Transaction": "C",
  "Certification Time": "2021-03-01 14:46:23.016736",
  "Declarant ID": "XNJ6BMMQ0J2O0DCN3ZTICLBTAYSC58LC",
  "Signature": "UC86DHXNM5I26NUNJ9EJ9VKTSQHNLMAI",
  "Certification ID": 2,
  "Process ID": "1",
  "Conditions": {
    "Condition 1": "True",
    "Condition 2": "True",
    "Condition 3": "True"
  },
  "Documents": {
    "Document 1":
"owiHPXx3LFfU1QCTSB7272hp23dLVKOpt9iHuXTyvtXVVwRapVpx8BSYT76fPm4y
QTFjLnorcw4UqB93GyuwheIGT3cvzp5SG8u9",
    "Document 2":
"EPXeAI8rJ12YhCRKDjDhiIGPC7RiM5MzuY282W129weiNDQNCFmGnxjpsDu5RXdl
KJ8oyWeIFulVFIsyQAvWj9lO7s4UrtmXl1H0"
  }
}

{
  "Timestamp": "2021-03-01 14:46:23.016736",
  "Transaction ID": 1,
  "Kind of Transaction": "C",
  "Certification Time": "2021-03-01 14:46:23.016736",
  "Declarant ID": "DT3DM584INUV1DEUTNKQ5HOXQIP17AA2",
```

```
  "Signature": "FDNLWQW1KR2YY7LWHP51QPQDGKKMLRIS",
  "Certification ID": 1,
  "Process ID": "1",
  "Conditions": {
    "Condition 1": "True",
    "Condition 2": "True",
    "Condition 3": "True",
    "Condition 4": "True"
  },
  "Documents": {
    "Document 1":
"UM0iDUhsHfQGgc0uVHiIkgIvD1rD0RKtdPs71I5zwV7B5mEhJk5StZA4v16TCnqYq0
RFccTIiz3guB2iFC9sX624vlhb68KWChl2"
  }
}

{
  "Timestamp": "2021-03-01 14:46:24.016736",
  "Transaction ID": 2,
  "Kind of Transaction": "C",
  "Certification Time": "2021-03-01 14:46:24.016736",
  "Declarant ID": "SW1WSP9OM6CMAM5CU2NOZ2VUDTRMTLPL",
  "Signature": "SXBZ9LL0WQYGFMX39WHTMT47N6S4M18U",
  "Certification ID": 3,
  "Process ID": "1",
  "Conditions": {
    "Condition 1": "True",
    "Condition 2": "True",
    "Condition 3": "True",
    "Condition 4": "True",
    "Condition 5": "True",
    "Condition 6": "True",
    "Condition 7": "True"
  },
```

```
  "Documents": {
    "Document 1":
"QJKXiqKyB8B1LTtuoKTaMVuWkLwzXwJcqUlEiuog7dpPlmAgnR3wOsnbTR1ig0tsN
73jNPzfwenEjysTUzjDnrHCBMder0Jc98Eo",
    "Document 2":
"NqQS7axqyl7NAkeCFzKtoN3p6Okjq581UCOFRNgYLh2DYueUA5plr6v4oowiHbH8ef
YrWu928jiITbXS7Kz9w6w5bXZHQpRvs5Kd",
    "Document 3":
"oIcinP9lHxbeKmjNzciGYNW1K7CSaXbwI39KwJ2cdsbEX50XX4O4YJSU8LcoqTvd6
oSApQ4d3QbOpayto2DFtopNMqy1pQYpIH3a",
    "Document 4":
"kkJudKj9GO31hc4rDiejnn5ffWokDXK7j4QWp4sshMfU0g9T8w6fKekHbkGl7CUAcg9
EkOpT6Da2T7k2D8STJQXiWLFIoeaY9Zjm",
    "Document 5":
"g5Qbi86fJojTmmxorJuff1yoQhJjymKBHe1ELKtaYJFL5oCt79MeBBhoDDh59hVsRfTl
NbHNsLLpolluTQBk6T4YJt1C3aIN1Ypm",
    "Document 6":
"M6rGrbjMgrdqXtuhMci4qgTVK3iIS6kxerzbTcnA2tJoVj3cBLut2xZW5BLxqj8TaorMV
ZCoGevmOGABVa0HwgbvVd53N7tsFqVf"
  }
}
```


Nell'esempio sono rappresentate tre transazioni corrispondenti a tre diverse certificazioni ognuna delle quali composta da un singolo processo. Per questioni di spazio char è stato assunto pari a 100 e non 1000000. Nella prima transazione sono necessarie 3 condizioni verificabili su due documenti per il passaggio di processo, mentre nella terza si hanno 7 condizioni verificabili su 6 documenti.