POLITECNICO DI TORINO
Corso di Laurea Magistrale in Ingegneria Informatica

Tesi di Laurea Magistrale

Applicazione basata sulla tecnologia Blockchain per i processi di certificazione ISO

Relatore
Prof. Paolo Giaccone

Candidato
Stefano Suraci

Anno Accademico 2021-2022

Ringraziamenti

Volevo ringraziare il mio relatore accademico Professor Paolo Giaccone, per avermi seguito lungo il lavoro di tesi con precisione, professionalità e umanità.

I miei sinceri ringraziamenti sono rivolti anche al Dottorando Iman Lotfimahyari, per la sua pazienza e disponibilità. Il suo supporto è stato determinante per la mia formazione e per la mia crescita personale.

Un grazie di cuore anche ad Angela. In questi ultimi anni sei sempre stata al mio fianco ed è anche merito tuo se sono riuscito a raggiungere i miei obiettivi. Non potevo desiderare una compagna migliore.

Un pensiero va anche ai miei parenti. Avere la famiglia vicino è una fortuna rara e voi ci siete sempre stati. In particolare un grazie speciale a Fabio che da sempre è un punto di riferimento per me.

Vorrei ancora ringraziare i miei genitori che sono i miei maestri di vita. Mi avete dato la forza nei momenti più difficili e per questo vi sarò sempre riconoscente.

Infine, vorrei ricordare tutti gli amici e colleghi con cui ho condiviso gioie e fatiche in questi anni.

Sommario

Oggi molte imprese reputano la blockchain uno degli strumenti più rivoluzionari a loro disposizione: il business si basa sulle informazioni e tanto più queste sono accurate migliori sono le negoziazioni. La blockchain si presenta come un sistema neutro e affidabile, in grado di colmare il *gap* di fiducia che attualmente è garantito dalle grandi corporazioni centralizzate. L'introduzione del sistema blockchain nei processi di certificazione permette di aumentare la fiducia nello strumento di accreditamento: le relazioni tra soggetto accreditato ed ente di certificazione e la valenza internazionale dei certificati vanterebbero le proprietà di veridicità e immutabilità dei dati memorizzati. Da queste premesse ha avuto inizio il seguente lavoro di tesi che da un lato approfondisce lo studio della tecnologia blockchain e dei processi di certificazione ISO, dall'altro propone una sua applicazione attraverso l'uso della piattaforma Hyperledger Fabric. Questo *framework* permette di creare blockchain di tipo *permissioned* particolarmente interessanti per casi d'uso aziendali. Infatti, in questo contesto, è importante poter effettuare un controllo degli accessi alla rete, mantenere riservate alcune informazioni e riconoscere in ogni momento chi sta effettuando una transazione. A supporto dell'applicativo sono stati poi sviluppati altri due programmi: il primo permette di generare un file sintatticamente compatibile con il software Graphviz, uno strumento in grado di rappresentare informazioni strutturate come grafi; il secondo utilizza le informazioni fornite da Graphviz per generare automaticamente il codice relativo a un *chaincode*. Questi due strumenti permettono di adattare l'applicazione alle diverse esigenze aziendali eliminando la necessità di particolari competenze tecniche. Il lavoro svolto ha dimostrato l'effettiva possibilità di integrare la blockchain al contesto delle certificazioni, ma anche la necessità di strumenti e personale qualificato per regolamentare i contratti intelligenti.

Indice

Capitolo 1

Introduzione

1.1 Introduzione

La Blockchain rappresenta una delle tecnologie più innovative dell'ultimo decennio capace in poco tempo di catturare l'immaginazione di tantissime persone. In breve, permette la registrazione permanente, sequenziale e indefinita delle transazioni senza la possibilità di poterle cancellare. La sua nascita è formalmente riconosciuta nel 2009 con la creazione del Bitcoin, un sistema di pagamento elettronico Peer-to-Peer [5], da parte dell'anonimo fondatore Satoshi Nakamoto. Nella rete di Bitcoin i partecipanti possono scambiarsi la moneta digitale affidandosi esclusivamente all'architettura e al suo protocollo senza la garanzia di un autorità esterna. Questa descrizione apparentemente semplice ha delle implicazioni politiche e sociali non indifferenti: la blockchain ha le potenzialità di svolgere quel ruolo di fiducia che oggi è completamente in mano alle grandi corporazioni centralizzate.

All'inizio l'interesse degli sviluppatori si è concentrato principalmente in ambito finanziario, complice la popolarità del Bitcoin che è senza dubbio la sua applicazione più famosa. Le blockchain però consentono lo scambio di qualsiasi tipo di asset o valore e per quest, con il tempo, l'interesse verso questa tecnologia è aumentato notevolmente. Oggi, infatti, molte imprese lo reputano uno degli strumenti più rivoluzionari a loro disposizione: il business si basa sulle informazioni e tanto più queste sono accurate migliori sono le negoziazioni.

Da queste premesse ha avuto inizio il seguente lavoro di tesi che da un lato approfondisce la tecnologia blockchain e l'ambito delle certificazioni, dall'altro propone una sua applicazione attraverso l'uso della piattaforma Hyperledger Fabric. Al giorno d'oggi infatti, fornitori, investitori e clienti richiedono alle aziende un'attenzione sempre maggiore in merito a tematiche come l'ambiente, la qualità, la sicurezza e la sostenibilità. Pertanto, affinché un'azienda possa mostrarsi all'altezza di questi standard e ottenere un vantaggio competitivo, è necessario che ottenga un certificato aziendale. L'introduzione del sistema blockchain nel processo di certificazione permetterebbe di aumentare la fiducia nello strumento di accreditamento stesso. Gli accordi commerciali, le relazioni tra soggetto accreditato ed ente di certificazione, ma anche la valenza internazionale dei

certificati potrebbero vantare le proprietà di veridicità e immutabilità dei dati memorizzati. Anche l'azienda potrà trarre vantaggio da questa situazione testimoniando a tutti gli *stakeholders* l'effettiva qualità dell'organizzazione.

Infine, si sottolinea che il progetto è stato sviluppando su richiesta dell'organizzazione Bechain, una società che si occupa di accompagnare le aziende nel processo di transizione digitale. L'introduzione di Bechain, come organizzazione fornitrice del servizio ha influenzato, le scelte progettuali.

1.2 Punti chiave della tesi

Il seguente documento è suddiviso in due parti: nella prima parte si introducono i sistemi di riferimento e la piattaforma di sviluppo, nella seconda parte i risultati sperimentali.

Nel capitolo due si effettua uno studio approfondito del sistema Blockchain e del modello delle certificazioni. Nel primo caso si identificando i componenti principali, le tipologie e le principali differenze con i database distribuiti; nel secondo caso l'analisi è divisa in due parti: inizialmente si individuano le caratteristiche del sistema e si discutono i benefici della blockchain nell'ambito degli accreditamenti, successivamente si definisce un modello generale per i processi di certificazione.

Nel capito tre si introduce la piattaforma di sviluppo Hyperledger Fabric. Inizialmente si indicano le motivazioni che hanno condotto alla scelta di questo *framework*, successivamente si elencano le caratteristiche della piattaforma e le personali scelte implementative.

Nel capitolo quattro vengono introdotti due strumenti a supporto dell'applicazione. Anche in questo caso si giustifica la loro introduzione e si elencano gli obiettivi e le funzionalità. Infine, viene mostrato graficamente il loro utilizzo.

Nel capitolo cinque viene presentato, passo dopo passo, il funzionamento dell'applicazione. L'ordine delle operazioni ha come scopo la descrizione funzionale del programma e la dimostrazione del suo corretto svolgimento.

Infine, nel capitolo sei, si traggono le conclusioni e le possibili espansioni future.

Capitolo 2

Modelli di riferimento

2.1 Il sistema blockchain

La blockchain è un registro digitale, decentralizzato e distribuito su una rete di nodi nella quale un elenco di transazioni organizzate come sequenza di blocchi si occupano dell'archiviazione di dati. Ciascun blocco è legato al precedente da una funzione matematica, pertanto l'unica operazione applicabile sulla catena è l'inserimento di nuovi blocchi d'informazione alla fine del registro. L'ambiente in cui opera la blockchain è un ambiente in cui è assente la fiducia tra i diversi partecipanti e ciò che garantisce l'immutabilità e la sicurezza all'interno dell'ecosistema stesso è la crittografia e i protocolli di consenso che regolano l'interazione tra i nodi.

Tre aspetti fondamentali di una rete blockchain sono la decentralizzazione, la logica centralizzata e l'autorità decentralizzata [2]:

1. **Decentralizzazione**. La blockchain è un registro digitale decentralizzato e come tale le risorse sono distribuite e replicate nei nodi della rete. Pertanto l'applicazione viene eseguita da tutti i suoi partecipanti senza generare un singolo punto di possibile fallimento infrastrutturale.

2. **Logica centralizzata**. Per funzionare correttamente, una rete logicamente centralizzata deve essere determinata in ogni momento da un unico stato. Pertanto è necessario che tutti i nodi concordino sullo stato finale del sistema.

3. **Autorità decentralizzata**. In una rete con autorità decentralizzata non esiste un organo centrale pertanto tutti i nodi hanno gli stessi diritti. Segue che nessuno può impedire ad altri partecipanti di svolgere azioni sulla rete o forzare la censura di determinati contenuti.

2.1.1 I componenti della blockchain

Tutte le caratteristiche appena elencate sono il risultato di un meccanismo che si compone di diversi elementi [2]. I componenti e la loro funzione all'interno dell'ecosistema sono elencati nei paragrafi successivi.

Ledger

Il *ledger*, anche detto registro digitale, ricorda in molti suoi aspetti il registro contabile del libro mastro tradizionale. Nello specifico il concetto di *ledger* è associato a quello di registrazione delle transazioni e quest'ultime possono riguardare qualsiasi tipo di *asset*. Ad esempio il Bitcoin è associato al concetto di riserva di valore, ma le possibilità non si limitano a questo: un *asset* potrebbe essere un bene fisico, una proprietà, ma anche un bene intangibile come nel caso di una identità. Appare quindi evidente la similitudine tra *ledger* e database: entrambe le tecnologie si basano sull'idea di salvare delle informazioni, ma mentre in un database è possibile inserire, cancellare e modificare i dati, in un *ledger* è possibile unicamente aggiungere nuove informazioni, questa funzionalità viene definita *append-only*. A rendere fattibile tutto questo in una blockchain sono diversi fattori come la crittografia, la teoria dei giochi e la decentralizzazione.

Blocchi

Le transazioni sono organizzate in blocchi. Ciascun blocco è collocato uno di seguito all'altro e contiene una prova matematica calcolata attraverso l'uso di strumenti crittografici che ne garantiscono la disposizione in sequenza.

Funzione di hash

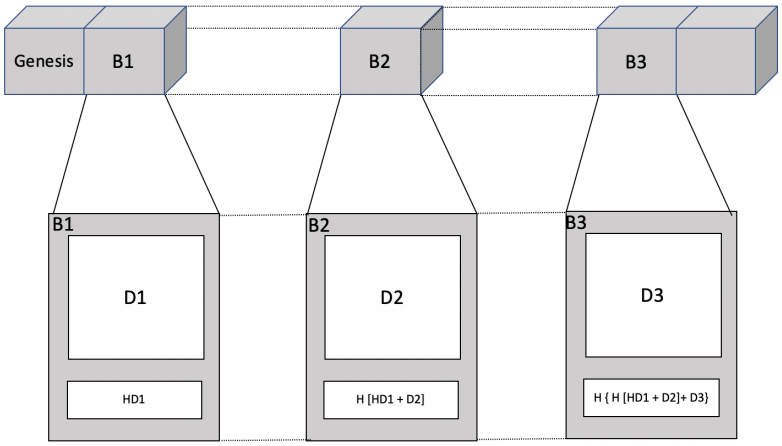

Figura 2.1. Catena di blocchi in una blockchain.

Si tratta di una funzione matematica utilizzata per associare dati di dimensione qualsiasi a dati di dimensione fissa. Questo significa che l'input di una funzione di hash potrebbe essere anche questo elaborato di tesi, ma l'output, chiamato "hash", sarà sempre caratterizzato da un numero di bit finito. Inoltre, la funzione di hash è una funzione deterministica: se fornissimo sempre lo stesso input, produrremmo sempre lo stesso output, mentre se fornissimo input anche leggermente diversi, produrremmo hash completamente

diversi. Un'ultima considerazione riguarda l'unidirezionalità delle funzioni di hash: calcolare un hash è molto semplice, l'operazione inversa è praticamente impossibile. In questo caso infatti l'unica strategia è quella di provare tutte le possibili combinazioni tramite un metodo a forza bruta (o di ricerca esaustiva).

La blockchain è letteralmente una catena di blocchi dove ciascun blocco contiene al suo interno una parte di dati, identificate come transazioni, e l'hash del blocco precedente. Per questa ragione, ogni volta che calcoliamo l'hash di un nuovo blocco, il blocco stesso è legato tramite la crittografia a quello precedente poiché il suo hash è parte integrante. Di conseguenza, se un utente malevole provasse a modificare o ad aggiungere una o più informazioni all'interno di un qualsiasi blocco, andrebbe inevitabilmente a cambiare il suo hash. Ed è proprio questo meccanismo che giustifica l'utilizzo diffuso della funzione di hash nei sistemi blockchain, in quanto ogni tentativo di manomissione risulterebbe immediatamente evidente. Inoltre, la valutazione dello stato corrente, richiede solo l'analisi dell'ultimo blocco e questo facilita la verifica.

Algoritmo di Consenso

La blockchain è un ambiente in cui è assente la fiducia tra i diversi partecipanti, ed è in questa incertezza che i nodi, attraverso regole matematiche, devono giungere a un accordo sullo stato della blockchain. La blockchain infatti, a differenza dei sistemi centralizzati, non ha un governante e solo la definizione di un processo di consenso permette a tutti i nodi di prendere decisioni comuni. Possiamo pensare al consenso come a un accordo basato su regole tra i diversi partecipanti della rete blockchain ognuno dei quali ha un proprio ruolo.

Segue che in un sistema blockchain raggiungere il consenso non è banale e il problema dei generali bizantini evidenzia questa difficoltà [12]. Il dilemma vede protagonisti un gruppo di generali disposti in diversi luoghi della città e che devono prendere una decisione unanime riguardo la possibilità di attaccare o di ritirarsi dalla città. In questo situazione bisogna fronteggiare il problema che i generali possono mentire riguardo la loro decisione e che i messaggeri, responsabili di recapitare i messaggio tra i diversi generali, possono smarrire o distruggere gli stessi, arrivare in ritardo, o essere anch'essi mendaci.

Se applichiamo il dilemma al contesto della blockchain, ciascun generale rappresenta un nodo della rete e l'unico modo per raggiungere il consenso è avere la maggior parte dei nodi della rete affidabili e onesti. Nel caso in cui questo obiettivo non venga raggiunto, la rete è vulnerabile a un possibile attacco di tipo *cinquantuno per cento*. Questo tipo di attacco si verifica quando un gruppo, ma anche un singolo, controlla più della metà della potenza di calcolo della rete.

Il consenso in una rete blockchain viene raggiunto tramite il *mining*. Con il termine minare ci si riferisce unicamente alla procedura composta dalla convalida delle transazioni, dal loro inserimento all'interno di blocchi e dall'aggiunta dei blocchi alla blockchain. Esistono diverse algoritmi in grado di svolgere queste operazioni risolvendo allo stesso tempo il problema dei generali bizantini; uno fra questi è il *Proof of Work*. La prova di lavoro è un problema matematico in cui diversi nodi della rete competono per trovare una soluzione. Il problema è molto simile alla ricerca del valore corrispondenza ad funzione di hash, per questo motivo il metodo di risoluzione si basa sulla ricerca esaustiva: il nodo

che per primo risolve il problema viene ricompensato per il suo lavoro e crea un nuovo blocco.

Crittografia a chiave pubblica

La crittografia a chiave pubblica è ampiamente utilizzata in internet e ricopre un ruolo di primaria importanza anche in molte dinamiche che coinvolgono la blockchain. Questo meccanismo è utilizzato per la generazione degli indirizzi sulla blockchain e l'autenticazione delle transazioni tramite la firma digitale.

L'idea alla base della crittografia a chiave pubblica è l'utilizzo di una coppia di chiavi che abbiano tra di loro una relazione matematica, in particolare:

- la chiave privata è generata casualmente e deve essere conosciuta solo da chi la possiede;
- la chiave pubblica è generata dalla chiave precedente attraverso una funzione matematica. In questo caso la chiave può essere condivisa.

Entrambe le chiavi sono dei numeri esadecimali molto grandi. Generare una chiave pubblica partendo da una chiave privata non richiede molta potenza di calcolo, mentre l'operazione inversa è impossibile da svolgere in un tempo ragionevole, in quanto richiederebbe milioni di anni.

Le firme digitali servono per dimostrare l'identità di una persona nel momento in cui questa non sia presente fisicamente; esattamente come le firme autografe. L'unica differenza è che al posto dell'utente che scrive la sua firma manualmente, vengano utilizzate funzioni matematiche. Le firme digitali sono il risultato della combinazione di funzioni di hash e crittografia a chiave pubblica e permettono di garantire diverse proprietà che in un ambiente non sicuro come quello di internet risultano fondamentali. Precisamente queste proprietà sono la criptazione, l'autenticazione, l'integrità e il non-ripudio.

Come accennato in precedenza la blockchain è pubblica, ma su di essa non sono definiti dei profili utente, bensì degli indirizzi. Gli indirizzi in quanto tali rappresentano la destinazione di una transazione e, attenzione, non contengono *asset*: la blockchain è un registro.

L'indirizzo sulla blockchain è il risultato di algoritmi crittografici basati sulle funzioni di hash che hanno come parametro di input la chiave pubblica. Questi indirizzi, in quanto destinazioni, sono solitamente utilizzati con strumenti chiamati *wallet*.

Wallet

Un *wallet* è un portafoglio, ma a differenza del classico portafoglio non contiene denaro. Il ruolo di un *wallet* è quello di memorizzare la chiave pubblica e la chiave privata di un indirizzo e tramite esso accedere ai dati che sono memorizzati sulla blockchain. Questo spiega ancora una volta le ragioni per cui la chiave privata non deve essere ne condivisa e ne persa, in quanto costituisce l'unico modo per perdere l'accesso alle risorse. Le risorse infatti resteranno sempre memorizzate sulla rete blockchain a prescindere dallo smarrimento della chiave stessa.

Transazioni

Un *ledger* è un elenco di transazioni e una transazione valida è l'informazione minima che viene scritta sul registro digitale.

Ad ogni nuova transazione valida lo stato nella blockchain cambia come evidenziato dal principio di logica centralizzata. Un esempio pratico è rappresentato dal conto bancario in cui l'elenco delle transazioni rappresenta il movimento del saldo, mentre il saldo finale è lo stato attuale della blockchain. E' bene sottolineare ancora una volta che l'*asset* può essere qualsiasi cosa, non necessariamente monetario come nel caso di Bitcoin.

Fino a questo momento abbiamo parlato di transazioni valide in quanto una transazione può assumere due stati:

- valida, modifica lo stato della blockchain;
- non valida, lo stato della blockchain rimane invariato.

Pertanto, lo spazio dei possibili stati non include nessuno stato intermedio ed è per questo che le transazioni sono deterministiche, ossia non possono essere rifiutate o modificate. In questa circostanza vediamo la grande differenza con le transazioni bancarie: una transazione valida è immutabile. Come verrà spiegato nel paragrafo successivo, gli *smart contracts* gestiscono questa situazione, permettendo di aggiungere conferme ulteriori prima che una transazione venga effettivamente considerata valida e quindi aggiunta definitivamente alla blockchain.

Smart Contract

Gli elementi alla base di ogni negoziazione sono i contratti. E' proprio grazie al contratto che si creano le condizioni di fiducia in una transazione di business. Questo è vero sia quando acquistiamo il biglietto per andare a teatro, che in situazioni molto meno frequenti come la stipula di un contratto di mutuo. Ciò che notiamo in entrambi gli esempi è che il contratto è un accordo tra le parti con valore legale. E' in questo contesto che si pone l'obiettivo di uno *smart contract* ossia assolvere in modo automatizzato le condizioni del contratto, prevenendo la possibilità di azioni non corrette da parte delle entità coinvolte. Questa è la ragione per cui l'idea di *smart contract* si inserisce perfettamente nel contesto della blockchain, un ambiente in cui la fiducia è parte integrante della tecnologia stessa.

Dal punto di vista pratico uno *smart contract* non è altro che un programma arbitrario eseguito sulla rete. La logica di queste applicazioni modella le caratteristiche di un contratto reale e il linguaggio di programmazione con il quale è scritto è per definizione privo d'incertezze. Per questo motivo l'accordo non è dipendente da leggi esterne e non è necessaria un'autorità che faccia da garante per farle rispettare. Sono infatti sufficienti il contratto stesso e il consenso della rete.

Solo quando le condizioni del contratto sono soddisfatte, lo *smart contract* esegue autonomamente le azioni per cui è stato programmato. Nel caso di Bitcoin è il trasferimento di denaro.

2.1.2 Tipologie di blockchain

Distinguiamo due tipologie di blockchain:

1. Blockchain pubbliche o *permissionless*. Le blockchain pubbliche sono attualmente le più utilizzate e il caso Bitcoin è l'esempio più evidente. In questa tipologia di reti non c'è un'autorità unica, ma bensì tanti singoli nodi tutti con gli stessi diritti e chiunque può unirsi alla rete senza possibilità di essere rifiutato. Nella rete non si fanno distinzioni in base alla provenienza o alla destinazione e per questo il sistema è per definizione aperto, resistente alla censura e assolutamente neutrale.

   Un sistema aperto ha anche altri vantaggi: in primo luogo permette di esplorare e verificare ogni transazione in quanto pubblica, in secondo luogo il codice sorgete è molto spesso *open source*. La condivisione del codice permette a chiunque da un lato di segnalare eventuali bachi, dall'altro di suggerire modifiche o nuove funzionalità.

2. Blockchain private o *permissioned*. In alcuni contenti come quello aziendale le caratteristiche della blockchain pubblica potrebbero non essere la soluzione migliore. A differenza delle blockchain pubbliche completamente aperte e prive di censura, quelle private introducono alcune differenze. I principali aspetti che caratterizzano il modello privato e che si differenziano da quello pubblico sono i seguenti:

   - controllo degli accessi alla rete generalmente su invito;
   - regole su chi può leggere e scrivere dati ed effettuare transazioni sulla blockchain;
   - riconoscimento di chi effettua una transazione;
   - possibilità di mantenere riservate alcune informazioni;
   - regole su chi partecipa al processo di consenso;
   - migliori prestazioni.

   Queste differenze sono il principale motivo per cui il modello privato risulta generalmente più favorevole per le aziende. Quest'ultime infatti instaurano spesso tra loro delle collaborazione in cui la conoscenza dei partecipanti e la confidenzialità delle informazioni sono un requisito fondamentale.

   Un altro motivo per cui è importante conoscere i nodi della rete è l'ambito delle regolamentazioni: purtroppo ad oggi questo ostacolo è ciò che maggiormente impedisce l'accettazione delle criptovalute.

   Per queste sue peculiarità la blockchain privata è un sistema che si colloca a metà tra la blockchain pubblica e il sistema centralizzato.

Confrontando i due modelli si può affermare che non esiste un sistema migliore dell'altro, ma la scelta dipende esclusivamente dal tipo di scenario.

2.1.3 Confronto tra blockchain privata e database distribuito

Molto spesso si colgono delle similitudini tra le blockchain e i database distribuiti. Entrambi sono dei sistemi in grado di memorizzare su diversi nodi o entità della rete informazioni in modo distribuito. Nonostante l'evidente analogia come sistemi in grado di gestire transazioni di dati, ci sono alcune caratteristiche che oltre a differenziali, li rende preferibili in contesti diversi. In generale è preferibile usare le blockchain quando le applicazioni devono essere eseguite in ambienti non affidabili. A questo proposito non è raro il caso di database invalidati a causa di attacchi da parte di utenti malevoli. Al contrario, quando i requisiti richiedono maggiori performance, i database distribuiti costituiscono la scelta migliore. In questo caso il protocollo delle blockchain rappresenta il principale collo di bottiglia.

E' possibile approfondire lo studio dei due sistemi analizzando tre dimensioni progettuali diverse [6]:

1. Replica dei dati. Nel caso dei database distribuiti i nodi non conoscono la logica delle transazioni e pertanto le valutano ed eseguono in modo indipendente. L'entità che gestisce le transazioni oltre a dover essere considerata affidabile è l'unica che si occupa di rendere disponibili i dati secondo criteri e regole che favoriscano le performance. Diversamente nel caso delle blockchain, non essendoci l'autorità fidata, è necessario che ogni nodo replichi l'esecuzione dell'intera transazione pagando un costo sia in termini di traffico sulla rete, sia in termini di *overhead* dovuto alle repliche del *ledger* su ciascun nodo. Inoltre, per garantire la verificabilità delle transazioni, nelle blockchain è necessario mantenere anche altre informazioni aggiuntive come ad esempio il *timestamp*, la firma digitale e il contesto.

2. Concorrenza. In un database distribuito si usano dei meccanismi per avvantaggiare l'esecuzione in parallelo delle transazioni. La concorrenza, infatti, costituisce il fattore che più incide sulle prestazioni. Al contrario, in una blockchain, le transazioni vengono eseguite in modo sequenziale poiché questo favorisce l'esecuzione deterministica degli *smart contracts* e la correttezza dello stato finale del *ledger*. Nonostante queste considerazioni, è necessario sottolineare che altri componenti della blockchain come il protocollo di consenso hanno un impatto decisamente maggiore rispetto all'esecuzione sequenziale. Per tali ragioni la differenza di performance rispetto a questa dimensione può passare in secondo piano.

3. Archiviazione. I database distribuiti mantengono solo lo stato finale del sistema, mentre i dati storici vengono memorizzati solo parzialmente poiché il loro scopo è sostanzialmente garantire un punto di ripristino in caso di errore. A questo proposito si sottolinea che sono permesse, oltre all'operazione di aggiunta, anche quella di modifica e di cancellazione. Diversamente nelle blockchain, oltre allo stato finale del sistema, è mantenuto lo storico di tutte le transazioni precedenti. In questo caso non è possibile tornare indietro o effettuare cancellazioni poiché l'intera catena è protetta da hash. Pertanto la blockchain richiede un costo maggiore in termini di memoria occupata.

2.1.4 Considerazioni

La blockchain si presenta come un sistema neutro e affidabile in grado di spostare la fiducia dai singoli individui o dalle istituzioni, alla tecnologia e alle sue applicazioni. A questo proposito le *macchine* risultano agevolate: un nodo esegue meccanicamente le istruzioni per cui è stato programmato, indipendentemente dalle scelte degli altri partecipanti. Un sistema di questo tipo non è perciò condizionato da eventi esterni come la corruzione o la difficoltà di compiere scelte eque tipiche degli esseri umani. La blockchain, quindi, non rappresenta solamente un periodo di transizione tecnologico poiché cambiano completamente gli schemi: i modelli centralizzati e chiusi lasciano spazio alla decentralizzazione all'*open source*.

2.2 Il modello delle certificazioni

2.2.1 ISO

International Organization for Standardization (ISO)

L'ISO è un organizzazione autonoma che opera nel mercato internazionale ed è costituita da un team di esperti il cui scopo è creare standard validi in tutto il mondo. Ciò che rende possibile la creazione di queste norme è la condivisione di conoscenze da parte dei membri costituenti, la cui provenienza spazia in ben oltre 150 istituzioni nazionali [3].

L'ISO opera in un contesto in continua evoluzione, il mondo di oggi è molto diverso da quello passato e lo sarà anche in futuro. Per questa ragione lo scambio d'idee e la cooperazione sono molto importanti nell'ottica del continuo miglioramento. Per garantire standard di qualità, ISO ha individuato le quattro macro aree che più influenzano la definizione di norme internazionali, questi settori sono la tecnologia, l'economia, l'ambiente e la società. Le innovazioni tecnologiche hanno un forte impatto sulla società in quanto da un lato permettono di creare nuovi lavori, dall'altro possono migliorare la produttività di quelli già esistenti. Gli standard hanno il duplice compito di valutare la bontà delle tecnologie e di suggerire alle imprese quali di queste possono favorire il loro processo di transizione digitale. Dal punto di vista economico l'attuale sistema commerciale globale favorisce la domanda di standard internazionali. Anche la tematica ambientale è un tema molto sentito a livello mondiale. Per poter rispondere al cambiamento climatico è necessario uno sforzo comune internazionale e l'ISO con la definizione di standard che prescindono dai confini nazionali promuove la transizione sostenibile. Infine, le società moderne risultano sempre più attente in materia di trasparenza delle informazioni e di privacy. In questo contesto ISO si impegna alla realizzazione di norme che permettano alle aziende di rispondere alle richieste degli *stackholders*.

Risulta evidente che l'attività svolta da ISO ogni giorno oltre a essere molto nobile ha ripercussioni immediate nella vita quotidiana di tutti noi. Ciò che rende valida l'attività svolta da ISO è il consenso distribuito da parte di specialisti provenienti da tutto il mondo e quindi la fiducia riposta dagli stessi nei confronti dell'ISO come piattaforma imparziale di standardizzazione. Nella sua volontà di facilitare, migliorare e rendere più sicuri i processi, ISO si impegna su diversi fronti: per prima cosa si sforza di diffondere standard a sempre più persone e aziende. Gli standard, oltre a dover essere qualitativi e aggiornati, devono essere spiegati agli individui in modo che possano comprenderne i benefici. In questo senso solo il consenso generalizzato facilita la diffusione. Successivamente lo sforzo si rivolge a questioni interne all'organizzazione: per garantire la qualità del servizio offerto è importante ascoltare le parole di tanti specialisti favorendo l'accoglienza e la diversità.

Accredia

Come previsto dalle direttive Europee ciascun stato membro ha il proprio ente di accreditamento [8]. In Italia l'ente unico stabilito dal governo è Accredia che agisce secondo quanto statuito dalla norma ISO 17001. Il compito di queste istituzioni è assicurare che gli enti di certificazione siano idonei a valutare in modo obiettivo sistemi, servizi, prodotti

e persone. Pertanto Accredia, insieme agli altri enti europei, collabora per poter infondere fiducia sull'intero mercato e salvaguardare l'ambiente e le persone.

Ente di certificazione

La certificazione è un attestato rilasciato da un ente accreditato che soddisfa un insieme di norme stabilite a livello internazionale. L'obiettivo di un certificato è assicurare che il sistema, il servizio, il prodotto o il personale soddisfi i requisiti specificati nella certificazione stessa. Il loro soddisfacimento è fondamentale per infondere fiducia e tale credibilità non si limita alle controparti ma si estende pubblicamente. I certificati infatti hanno una valenza internazionale e l'ente di certificazione che li rilascia costituisce una terza parte imparziale durante tutto il processo di certificazione.

Tipologie di certificati

Esistono diverse tipi di certificazione: le certificazione di prodotto determinano l'idoneità di un bene prima della sua distribuzione sul mercato. Per alcuni prodotti questo tipo di certificazione è un requisito obbligatorio come nel caso dei dispositivi medici, ma per altri la scelta è volontaria. In questa tipologia di certificati rientrano anche i servizi e i processi produttivi: in questo caso la verifica da parte delle norme si focalizza sulla bontà della metodologia e dell'applicazione delle singole parti. Un'altra tipologia di certificazioni riguarda i sistemi di gestione. Le certificazioni in questo ambito analizzano l'apparato gestionale delle organizzazioni. Le imprese possono essere private o pubbliche e appartenere a un qualunque ambito. L'obiettivo di queste norme è verificare la validità delle aziende dal punto di vista della qualità, della sicurezza e della sostenibilità. Infine, distinguiamo le certificazioni del personale. Lo scopo di questo documento è duplice: da un lato assicurare le capacità e le competenze tecniche, dall'altro verificare il loro mantenimento: in questo modo gli specialisti sono motivati ad aggiornarsi e migliore continuamente.

Vantaggi

Le certificazioni possono essere volontarie o previste dalla legge. La certificazione ISO 9001 ad esempio non è obbligatoria ma è un requisito indispensabile per tutte quelle imprese che desiderano essere competitive sul mercato. Il possesso di questo certificato infatti aumenta le opportunità lavorative perché assicura un punteggio migliore ai bandi pubblici. A giovarne è anche l'immagine dell'azienda nei confronti dei clienti, dei fornitori e dei collaboratori. Ottenere un certificato non solo dimostra l'interesse verso tematiche d'interesse comune come l'ambiente, la sicurezza e la sostenibilità, ma anche l'effettiva capacità organizzativa e la volontà di trasparenza. Infine, la transizione verso standard internazionali consente di rimanere al passo con gli altri *competitors*, ridurre i costi delle assicurazioni, ottenere agevolazioni fiscali e rendere più efficienti i processi produttivi.

2.2.2 Certificazione ISO 9001

La definizione di un modello in grado di combinare le proprietà del sistema blockchain con le esigenze dei processi di certificazione, non può prescindere da uno studio più approfondito delle caratteristiche fondanti di un certificato e delle modalità attraverso cui un'organizzazione può ottenerlo. A titolo di esempio si è scelto di approfondire lo studio della certificazione ISO 9001 che, oggigiorno, risulta fondamentale per attestare se un azienda opera nell'ambito della qualità.

Qualità

Quando si parla di qualità non è possibile specificare caratteristiche assolute. La qualità è un processo in continua evoluzione risultato delle necessità e dei bisogni delle persone. Nel 2015 la norma ISO 9000 fornisce una definizione della qualità che non si sofferma sul prodotto o sul servizio, ma piuttosto la individua come uno scopo. La qualità viene quindi intesa come un modo di operare indispensabile per tutte quelle aziende che intendono essere efficienti e competitive sul mercato.

Sui principi alla base della norma ISO 9000 si fonda lo standard ISO 9001 che delinea le caratteristiche di un Sistema di Gestione per la Qualità. L'SGQ ha come obiettivo la realizzazione del cliente e per soddisfare le sue esigenze analizza le attività di gestione delle aziende. In particolare individua i settori che compongono l'organizzazione, il ruolo delle persone che lavorano al suo interno e gli obiettivi che entrambi si prefiggono. Infine analizza le procedure più importanti e come le attività sono programmate. Si noti che una volta progettato, l'SGQ deve essere continuamente implementato per garantire il miglioramento continuo e la risoluzione di eventuali non conformità. All'interno di una organizzazione un SGQ può individuare diversi livelli di maturità: basso, medio o alto. La maturità si misura sempre in riferimento al soddisfacimento dei clienti e infatti il paramento utilizzato considera il numero di reclami. Pertanto un livello di maturità basso determina un risvolto negativo dell'SGQ.

Requisiti

Per l'implementazione di un sistema di gestione per la qualità, la norma ISO 9001 individua dei principi generali di realizzazione e mantenimento che possono essere adottati da una qualsiasi tipo d'impresa. Pertanto le norme prescindono dal servizio e dal prodotto offerto, ma anche dalle dimensioni delle organizzazioni. L'unico requisito che viene richiesto alle aziende è la volontà di dimostrare l'efficacia del proprio sistema. Lo standard ISO 9001, a differenza della norma ISO 9000 che fornisce delle linee guida, può essere certificata dalle organizzazioni. Possedere tale certificato non è obbligatorio ma è un requisito indispensabile per tutte quelle aziende che, consapevoli della concorrenza, vogliono ottenere un vantaggio competitivo sul mercato.

Struttura

L'ISO 9001 è strutturato in dieci sezioni, le prime tre individuano lo scopo, il campo applicativo, le norme di riferimento e i termini e le condizioni, le successive sette identificano

i punti cardine di un SGQ. In questo ultimo caso distinguiamo l'ambito dell'organizzazione, la leadership, il supporto, la pianificazione, le attività operative e il miglioramento. Analizzandole più nel dettaglio. Il contesto dell'organizzazione ha come obiettivo definire lo scopo dell' SGQ sulla base dei processi aziendali e della capacità di un azienda di raggiungere i propri obiettivi. La leadership analizza le capacità della parte direttiva dell'azienda nel distribuire ruoli e responsabilità. La pianificazione riguarda la valutazione dei rischi e delle contromisure. Il supporto riguarda il monitoraggio di tutte le risorse dell'organizzazione, sia umane, che fisiche, che ambientali. Le attività operative valutano i requisiti e le prestazioni delle attività anche in termini di soddisfacimento del cliente e Audit interni. Il miglioramento valuta eventuali non conformità e contromisure.

Principi

I principi che regolano un Sistema di Gestione per la Qualità sono sette. Il primo principio prende in considerazione il cliente sottolineando l'importanza di mettere al centro il consumatore in tutte le fasi del processo aziendale. Il secondo principio si concentra sulla necessità di un leader capace e consapevole degli obiettivi della propria azienda. Il terzo principio pone l'attenzione sui lavoratori che compongono un'organizzazione e sull'importanza del loro coinvolgimento. Il quarto principio evidenzia l'importanza di ogni sotto processo nella realizzazione del risultato finale. Il quinto processo si collega a quello precedente analizzandolo dal punto di vista gestionale. Il sesto principio si focalizza sul valore scientifico delle scelte effettuate. L'ultimo principio è un invito a cercare di migliorarsi sempre.

Vantaggi

Un'organizzazione che decide d'introdurre un Sistema di Gestione per la Qualità ottiene diversi vantaggi. Per quanto riguarda i vantaggi interni, il primo valore aggiunto considera la riduzione dei costi e dei tempi di produzione. Successivamente si noti che l'SGQ migliora la capacità dell'organizzazione in materia di gestione dei processi e nell'identificazione e distribuzione di ruoli e responsabilità. I processi aziendali diventano sempre più trasparenti favorendo benefici esterni: il cliente risulta maggiormente soddisfatto e aumentano le possibilità fidelizzazione.

Processo di certificazione

Dopo aver identificato i requisiti, la struttura, i principi e i vantaggi relativi alla certificazione ISO 9001, in questo paragrafo elenchiamo quali sono i passaggi richiesti a un'organizzazione che intende certificarsi nell'ottica della qualità.

Il primo passo per un'organizzazione che intende ottenere un certificato ISO 9001 è prepararsi alle richieste previste dalla norma. Questo significa preparare la documentazione ed eventualmente organizzare *audit* interni. Gli *audit* interni sono eseguiti da membri interni all'azienda, specializzati nel verificare la corrispondenza tra quanto previsto dalla norma e quanto offerto dall'impresa. Non appena l'organizzazione ritiene di possedere tutti i requisiti elencati nello standard, contatta un ente di certificazione. L'ente di certificazione è un organizzazione accreditata che organizza gli *audit*. Essendo

l'organo di certificazione esterno in questo caso si parla di *audit* esterno. Alla richiesta di certificazione ricevuta dall'azienda, l'ente di certificazione risponde con un preventivo il cui costo dipende dal numero di dipendenti impiegati, dal tipo di certificazione e dalla quantità e complessità delle unità e dei processi aziendali. A questo punto l'azienda ha due possibilità: accettare o rifiutare l'offerta. Se l'offerta è rifiutata, bisogna ricominciare da capo, se l'offerta è accettata, si prosegue con la pianificazione dell'*audit* che definisce la data e l'ora dell'incontro. Successivamente ha inizio la fase di *audit* vera e propria che si divide in *stage 1* e *stage 2*. Nella prima parte si controlla la documentazione e si accerta l'effettiva organizzazione dell'azienda in vista della fase successiva. In caso di errori, è necessario risolverli prima di procedere alla fase successive. La seconda parte dell'*audit* prevede la verifica di tutti i requisiti di conformità previsti dallo standard; in questo caso il controllo è sempre effettuato sul luogo di lavoro. L'*audit* termina con una delle seguenti tre valutazioni: conforme, non conforme lieve o non conforme grave. Nel primo caso l'azienda ha superato le verifiche e riceve il certificato ISO. Nel secondo caso le non conformità possono essere risolte facilmente e si può programmare un nuovo *audit* in poco tempo. Nell'ultimo caso, invece, non è possibile procedere nel breve periodo perché le divergenze sono particolarmente profonde. Se l'azienda ottiene il certificato ISO 9001, la sua validità ha una durata di tre anni, al termine dei quali è necessario ripetere tutte l'iter. Inoltre, durante i tre anni di validità del certificato, devono essere effettuati dei controlli periodici detti *audit di sorveglianza* per verificare il mantenimento dello standard. Solitamente questi *audit* sono meno intrusivi dei precedenti e possono essere svolti sia dallo stesso ente di certificazione che da altri.

2.2.3 Applicazione della blockchain nei processi di certificazione

In questo paragrafo cerchiamo di analizzare in che modo la tecnologia blockchain potrebbe migliorare l'ambito delle certificazioni e in quale configurazione potrebbe essere integrato.

In merito a questo argomento Accredita ha già iniziato a esprimersi dimostrando come l'interesse verso questa tecnologia non stia passando inosservato [7]. Nel capitolo precedente abbiamo discusso i vantaggi che il possesso di un certificato potrebbe determinare nei confronti di prodotti, servizi, processi e persone. Ma ciò che rende valido un documento sono le informazioni contenute al suo interno. Al giorno d'oggi i certificati sono in alcuni casi ancora rilasciati in modo cartaceo rendendo la loro replica un problema davvero importante. In altri casi sono invece smaterializzati e trasportati all'interno di sistemi digitali. Il digitale introduce numerosi benefici, ma solo a patto che sia possibile garantire la sicurezza delle informazioni: nei sistemi informazioni attuali i certificati sono salvata all'interno dei database di ciascun ente di certificazione e accreditamento. Nonostante la bontà delle tecnologie moderne, la possibilità che l'output prodotto da questi sistemi possa essere un giorno compromesso è un ipotesi che non può mai essere del tutto esclusa.

A seguito di queste premesse introduciamo ora la tecnologia blockchain, ricordando alcune delle sue caratteristiche principali: la blockchain permette di memorizzare in modo immutabile e trasparente le informazioni. Risulta evidente che una delle applicazioni più immediate della blockchain potrebbe essere quella di includere le certificazioni all'interno di questo sistema. Accanto a questa ipotesi, un'altra idea potrebbe essere di aggiungere,

oltre al certificato, altre fasi del processo di certificazione, in particolare quelle in cui è richiesta una maggiore trasparenza dal mercato.

Immaginando d'implementare una di queste due strategie, cerchiamo adesso di comprendere in che modo istituzioni, imprese, clienti, fornitori e collaboratori potrebbero beneficiare di questa innovazione. Dal punto di vista delle istituzioni, la possibilità di memorizzare certificati, le cui informazioni sono garantite dalla tecnologia blockchain, permette di aumentare automaticamente la fiducia nello strumento di accreditamento. Lato imprese, il beneficio si potrebbe misurare in punteggi migliori ai bandi pubblici, in nuove opportunità lavorative e in agevolazioni fiscali e assicurative. Dal punto di vista dei clienti, dei fornitori e dei collaboratori, la trasparenza delle informazioni si tradurrebbe in maggiore soddisfazione.

Da queste considerazioni risulta evidente la possibilità di utilizzare la blockchain per ottenere un valore aggiunto nell'ambito delle certificazioni. Quando abbiamo introdotto la blockchain, è stata posta particolare attenzione sulla sua capacità di sostituirsi, in materia di fiducia, alle grandi corporazioni centralizzate. Nell'ambito delle certificazioni vediamo che questa tecnologia potrebbe seguire una strada diversa: attualmente la persona che fa da garante rispetto alle questioni legali di un contratto è il notaio. L'introduzione della blockchain e degli *smart contracts* introduce la necessità di figure che sappiano verificare l'idoneità dei contratti informatici e in generale dei processi che governano la blockchain. Pertanto non parliamo di completa sostituzione come nel caso di Bitcoin in materia di trasferimenti monetari, ma piuttosto d'integrazione. Risulta evidente che gli *smart contracts* possono autonomamente verificare difformità contrattuali più o meno gravi, ma allo stesso tempo enti di certificazione e accreditamento devono garantire attraverso personale sempre più competente e trasversale la loro efficacia.

2.2.4 Modello astratto

Il processo di certificazione ISO 9001 ha mostrato che alla base di dell'iter di certificazione vi è uno scambio di documenti tra un ente terzo e un cliente.

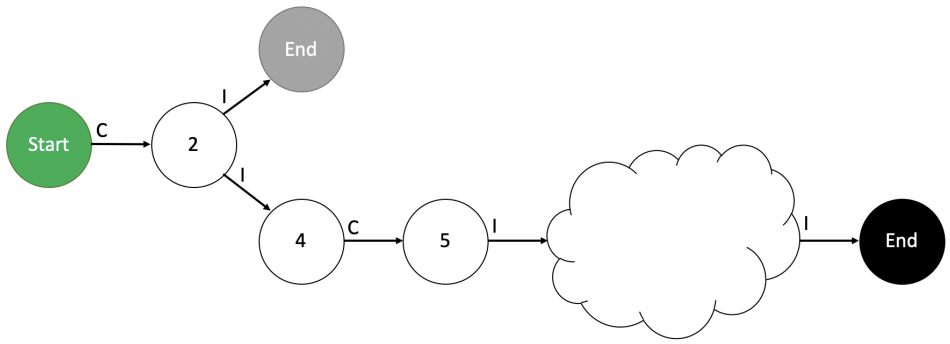

Figura 2.2. Esempio di macchina a stati generale per un processo di certificazione ISO.

L'immagine mostra che l'intero processo può essere modellato con una macchina a stati finiti aciclica deterministica, in cui il completamento di uno degli *step* del processo

di certificazione, comporta il passaggio da un stato i a uno stato successivo i+1. Gli archi in figura sono etichettati con le lettere C e I, cioè cliente e istituzione, a indicare quale attore deve svolgere la particolare transazione. Infine, la nuvola in figura, mostra che i processi di certificazione nonostante possano condividere alcuni stati, in altri casi differiscono a causa delle condizioni specifiche dei singoli contratti. Inoltre, anche il tempo e l'evoluzione delle norme può comportare delle modifiche.

Nel paragrafo relativo all'applicazione della blockchain nel contesto delle certificazioni, si era mostrata la possibilità d'include all'interno della blockchain le fasi principali dei processi di certificazione. Sulla base di queste osservazioni, tenendo in considerazione il processo di certificazione ISO 9001 e in riferimento all'automa a stati finiti, è stata realizzato un modello generale in grado d'includere gli aspetti principali di un qualsiasi processo di certificazione.

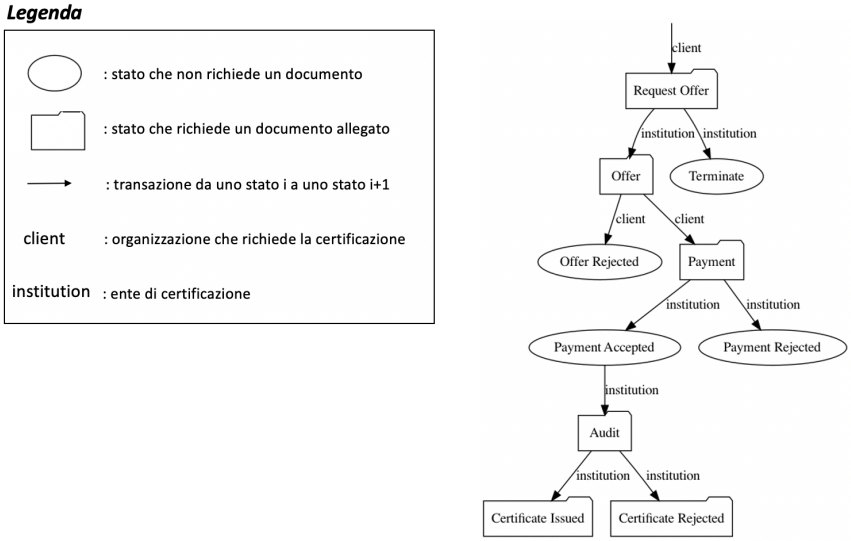

Figura 2.3. Modello astratto del processo di certificazione ISO generato con lo strumento *Graphviz*.

Prima di analizzare la macchina a stati in figura, descriviamo i simboli utilizzati per la rappresentazione. In modo simile al caso precedente, il *client* rappresenta un'organizzazione che richiede un certificato, mentre l'*istitution* è l'ente di certificazione. Le frecce indicano un passaggio da uno stato all'altro del modello astratto e rappresentano le transazioni effettuate o da un *client* o da una *istitution*. Gli stati rappresentati con la forma di ellisse indicano che la transazione non richiede un documento allegato, mentre quelle con la forma di una cartella lo richiedono.

Il processo di certificazione ha inizio nel momento in cui il cliente invia la documentazione di richiesta di offerta all'ente di certificazione. L'ente di certificazione può o rifiutare l'offerta o preparare il preventivo da mostrare al cliente. Nel primo caso non è necessario un documento, ma è sufficiente indicare che si declina la richiesta; nel secondo caso si invia il documento di *Offer*. Ricevuto il preventivo, il cliente può o rifiutare l'offerta o procedere con il pagamento. Lo stato *Payment* indica implicitamente che l'offerta è stata

accettata e richiede un documento da allegare alla transazione. Dal momento che i pagamenti sono una parte molto importante nelle negoziazioni, si è voluto dare particolare enfasi a questo processo. Per questa ragione, l'ente di certificazione deve segnalare la sua volontà di accettare o meno il pagamento: gli stati *Payment Accepted* e *Payment Rejected* non richiedono un documento. Se l'istituzione accetta il pagamento, si può procedere con la fase di *Audit*, che terminerà sempre su espressione dell'ente di certificazione nello stato *Certificate Issued* o *Certificate rejected*.

Risulta evidente che il modello astratto semplifica notevolmente il processo di certificazione, soprattutto se confrontato con l'esempio dell'ISO 9001. La scelta di ridurre drasticamente il numero di stati, è guidata dalla volontà di utilizzare solo quelli presenti in ogni processo di certificazione. Infatti, solo in questo modo è possibile definire un modello astratto generale. In ogni caso, come verrà mostrato nel capitolo quattro, parallelamente a questo modello viene creato uno strumento di supporto. Quest'ultimo è un programma che permette di generare una macchina a stati qualsiasi.

Capitolo 3

Piattaforma di sviluppo

3.1 Hyperledger Fabric

La capacità di garantire la fiducia decentralizzata su un sottoinsieme di entità note e la possibilità di mantenere confidenziali i dati e le transazioni sono solo alcuni degli aspetti che hanno portato a scegliere Hyperledger Fabric come piattaforma di sviluppo dell'applicazione.

Innanzitutto Hyperledger Fabric è una piattaforma ledger distribuita [11] e tra le principali caratteristiche differenziali bisogna menzionare che è stata creata dalla Linux Foundation. La Linux Foundation è un'organizzazione no-profit che vanta una comunità di sviluppo molto attiva e con un passato progettuale brillante. Recentemente anche IBM ha rinnovato il suo impegno per questa piattaforma sottolineando ancora una volta la sua validità.

Hyperledger Fabric si distingue per essere *open source* e per essere stata creato fin dall'inizio pensando alle aziende. Dal momento che le imprese possono implementare processi industriali differenti, Hyperledger Fabric offre un'architettura fortemente personalizzabile. Questa sua modularità la rende interessante a contesti aziendali molti diversi tra loro come ad esempio quello finanziario, sanitario, IoT, manifatturiero e musicale.

Un'altra caratteristica peculiare di Hyperledger Fabric è che permette agli sviluppatori di scrivere contratti intelligenti con linguaggi di programmazione noti. Si tratta infatti di linguaggi generici come Go e Java ampiamente conosciuti dalle aziende. Questo consente alle società di poter realizzare fin da subito le loro applicazioni eliminando il bisogno di formare o acquisire personale tecnico specifico.

Ciò che rende però Hyperledger Fabric davvero unico è la possibilità di definire protocolli di consenso altamente configurabili. La capacità di modellarsi è essenziale per adattare l'applicazione a schemi di fiducia eterogenei.

Per quanto riguarda i costi di realizzazione della rete, non essendo necessaria alcuna criptovaluta e di conseguenza nessuna operazione di *mining*, le spese sono paragonabili a quelle di un sistema distribuito generico.

Queste proprietà rendono Hyperledger Fabric una delle piattaforme blockchain *permissioned* più modulari e performanti e giustificano la scelta di utilizzare questa piattaforma.

3.2 Aspetti principali

Analizziamo ora con maggiore dettaglio gli aspetti principali che caratterizzano la blockchain di Hyperledger Fabric [10]:

Identità

Hyperledger Fabric permette di creare blockchain di tipo *permissioned*. Una rete di questo tipo richiede ai partecipante del *network* d'interagire con un'identità. A tal proposito la rete Fabric utilizza un'infrastruttura a chiave pubblica (PKI) caratterizzata da quattro componenti principali, quali: autorità di certificazione, certificato digitale, chiavi pubbliche e private ed elenchi di certificati revocati.

Le *certification authorities* (CAs) si occupano di distribuire certificati digitali alle diverse entità della rete tra cui i nodi *peer*, i nodi *orderer* e le applicazioni *client*. I certificati digitali da un lato identificano questi partecipanti, dall'altro determinano il loro accesso e autorizzazione alle risorse. Questo è possibile grazie alle chiavi pubblica e privata generate all'emissione del certificato, ma anche attraverso altre informazioni: nella rete Fabric i certificati sono emessi in conformità allo standard X.509 e includono attributi come il titolare del certificato, l'organizzazione e il ruolo. Inoltre le CA applicano la loro firma digitale al documento permettendo la verifica della validità di un attore nei confronti della sua chiave pubblica.

In alcuni casi è possibile che un certificato venga revocato e aggiunto alla *Certificate Revocation List* (CRL). In questo caso per non incorrere nel rischio di considerare valide le transazione di una entità pregiudicata, occorre verificare preventivamente la sua presenza nella CRL.

Membership Service Provider (MSP)

Introduciamo ora il concetto di *Membership Service Provider* (MSP) con un esempio: ogni volta che un nodo *peer* vuole firmare o approvare una transazione, applica la propria chiave privata. Per verificare la validità di questa firma, il nodo di ordinamento utilizza la chiave pubblica presente nell'MSP del nodo stesso. Quindi l'MSP è il sistema attraverso cui è possibile considerare una entità come affidabile.

Esistono due tipi di MSP: locale o di canale. Gli MSP locali risiedono localmente nei nodi *peer*, nei nodi *orderer* e negli applicativi *client* e determinano i permessi a livello di nodo. Nel caso del nodo *peer*, ad esempio, vorremmo poter definire chi sono gli amministratori del nodo o chi può effettuare l'installazione di un *chaincode* sullo stesso. Nel caso dell'applicazione *client*, invece, vorremo poter identificare l'utente che effettua la transazione come membro di un canale o come amministratore nel caso in cui la transazione sia di configurazione. Nel caso invece dei nodi di ordinamento vorremmo poter avere la lista dei nodi ritenuti affidabili dalla rete. L'Implementazione degli MSP locali avviene attraverso un insieme di cartelle definite unicamente nel *file system* del nodo ove le stesse si applicano. Al contrario, gli MSP di canale, definiscono la configurazione del canale, cioè le autorizzazione e le facoltà a questo livello. In questo caso ad esempio, è necessario che siano inclusi tutti gli MSP delle organizzazioni che fanno parte del canale

perché, altrimenti, le transazioni effettuate dai loro membri verranno scartate. Analogamente anche gli MSP delle organizzazioni che ospitano un nodo di ordinamento per quel canale devono essere incluse. Dal punto di vista implementativo, diversamente dal caso precedente, viene utilizzato un file con estensione json che è logicamente unico per il canale, anche se replicato fisicamente su ogni nodo. Segue che per la sincronizzazione delle repliche si utilizza l'algoritmo di consenso.

Politiche

Una delle principali differenze con le blockchain pubbliche e ciò che Hyperledger Fabric chiama *policy*. Nella blockchain di Bitcoin, ad esempio, uno qualunque dei nodi può generare o convalidare una transazione. In Fabric, al contrario, i partecipanti della rete sono noti e l'accesso alle risorse è determinato dalle politiche. In questo caso, una transazione per essere considerare valida deve essere confronta con la *governance*. Quest'ultima è decisa a priori, ma può essere modificata successivamente.

Hyperldger Fabric consente di stabilire *policy* in modo gerarchico. La configurazione *system channel*, ad esempio, permette alle entità di un consorzio di creare canali o, nel caso degli *orderers*, di stabilire come avviene il consenso e la generazione dei blocchi. Nel caso della configurazione *appliaction channel* il comportamento predefinito ereditata la configurazione del canale precedente, ma il funzionamento può essere cambiato. A questo livello si può stabilire chi aggiunge entità al canale, chi deve approvare la definizione di un *chaincode* e chi può accedere ai dati e ai contratti intelligenti. Infine, le liste di controllo degli accessi o ACL determinano l'accesso alle risorse e ai contratti intelligenti. Anche in questo caso il comportamento predefinito eredita i paramenti del canale precedente.

In Hyperledger Fabric le politiche possono essere approvate in modo esplicito o implicito. Nel primo caso parliamo di *signature policy* nel secondo di *ImplicitMeta policy*. Se ad esempio stiamo richiediamo la firma di un peer specifico dell'Org1 allora stiamo utilizzando una politica esplicita. Se, al contrario, richiediamo, ad esempio, l'approvazione della metà più uno dei partecipanti, allora siamo utilizzando una politica implicita. In questo ultimo caso la politica si adatta dinamicamente all'aggiunta o alla rimozione dei membri. Le politiche implicite combinano i risultati delle politiche esplicite e per questo si applicano implicitamente alle configurazioni di tipo *channel*.

Peers

Nei paragrafi precedenti abbiamo già menzionato i nodi *peer*, in questa sezione analizziamo con maggior dettaglio le loro funzionalità e il motivo per cui sono molto importanti nella rete di Hyperledger Fabric. I *peers* svolgono un ruolo centrale nel sistema perché da un lato ospitano i contratti intelligenti e le istanze dei *ledgers*, dall'altro permettono alle applicazioni di accedere a queste risorse. I tipi d'interazione possono essere di due tipi: interrogazione o aggiornamento. L'aggiornamento risulta più complesso poiché coinvolge il processo di consenso e quindi i nodi *peer* di tutte le organizzazioni; mentre l'interrogazione risulta meno onerosa perché può essere indirizzata unicamente al *peer* locale dell'organizzazione.

Nella rete di Fabric, i *peers*, in base alla funzione che devono svolgere, possono assumere ruoli diversi. Si noti che un ruolo non esclude l'altro e un nodo può svolgere anche tutte le funzioni. Con il termine *committing peers* ci si riferisce ai nodi del canale che durante la *validation phase* convalidano le transazioni nei blocchi secondo l'*endorsement policy* e i conflitti di serializzabilità. I *peers* di *endorsement* sono invece i nodi che firmano le transazioni quando vengono invocati dalle applicazioni *client*. Su questi nodi deve essere installato lo *smart contract*. Se un'organizzazione decide d'impiegare per ragioni di scalabilità e resilienza più di un nodo *peer* su un canale, un leader peer eletto staticamente o dinamicamente si occupa di inviare le transazioni dall'*orderer* ai *committing peers*. Un altro tipo di nodo è l'*orderer peer* che, insieme ad altri nodi dello stesso tipo, costituisce il servizio di ordinamento ed è responsabile di raggruppare le transazioni all'interno di un blocco. Infine gli *anchor peers* facilitano la comunicazione tra i *peers* di diverse organizzazioni.

Canale

In Hyperledger Fabric i canali creano un flusso di comunicazione privato tra le organizzazioni che li costituiscono. I membri della rete per poter effettuare transazioni devono unirsi ai canali con i propri nodi *peer*. Ciascun canale è caratterizzato da un proprio *ledger* e solo le organizzazioni costituenti possono accedervi in lettura e scrittura.

Dal momento che un'organizzazione può unirsi a uno o più canali, la rete Fabric permette ai suoi membri di partecipare nello stesso momento a diverse reti blockchain creando reti multiple all'interno della stessa infrastruttura. Questo consente da un lato di usare il network in modo più efficiente, dall'altro assicura la riservatezza dei dati e delle comunicazioni.

Ledger

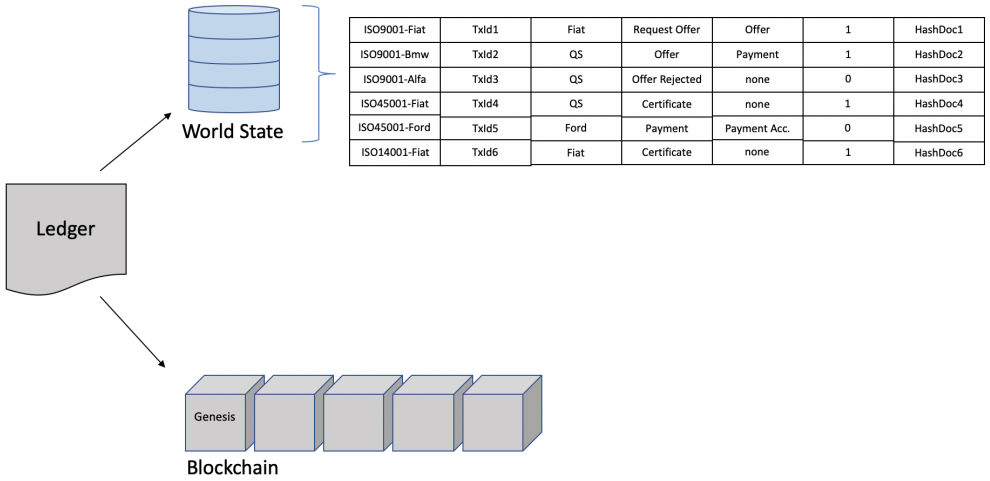

Figura 3.1. Ledger.

Il registro digitale è composto da due componenti strettamente legati tra loro, ma profondamente diversi nella funzione che svolgono e nella loro architettura: parliamo del *World State* e della Blockchain

Prima di elencare le caratteristiche e le differenze di ciascun elemento, riprendiamo l'esempio del conto bancario evocato nel capitolo riguardante la blockchain. Quando accediamo al nostro conto bancario la prima informazione immediatamente visibile è quella riguardante il saldo disponibile. Più in basso, solitamente, sono elencate le ultime transazione che abbiamo effettuato. Distinguiamo quindi due tipi di dato: da un lato il valore finale del saldo, dall'altro la lista delle transazioni. Possiamo pensare al *World Stato* come al componente che permette di accedere velocemente alla informazione del saldo disponibile, mentre alla blockchain come al registro delle transazioni che hanno condotto al valore corrente del *World State*.

In generale, su un *ledger*, possiamo memorizzare un qualsiasi tipo di oggetto aziendale. Più precisamente ciò che andiamo realmente a memorizzare non è l'oggetto in se, ma l'insieme delle informazioni che lo definiscono. In Hyperledger Fabric quest'ultime si chiamano *facts* e nello stato mondiale cambiano frequentemente. Ciò che invece non cambia ed è immutabile è la loro cronologia memorizzata sulla blockchain. Per queste ragioni le operazioni che si possono svolgere sui due componenti sono diverse: nel primo caso uno stato può essere aggiornato, eliminato o creato, mentre nel secondo caso può essere solo aggiunto.

Le differenti funzionalità e le differenti operazioni applicabili, giustificano la loro diversa implementazione. Nel caso del *World State* si utilizza un database perché fornisce un ampio insieme di operazioni per l'archiviazione e il recupero dei dati in modo efficiente. Al contrario, per la blockchain, si utilizza un file perché l'insieme delle operazioni è limitato principalmente all'aggiunta di uno stato alla fine del registro.

Smart Contract, Chaincode ed Endorsement Policy

Precedentemente abbiamo detto che il *ledger* contiene un insieme di fatti relativi a diversi oggetti aziendali. La logica attraverso cui i fatti degli oggetti aziendali sono aggiornati, eliminati o creati è contenuta all'interno degli *smart contracts*. I contratti intelligenti sono identificati univocamente dal nome del contratto e, nella rete Fabric, impacchettati all'interno dei cosiddetti *chaincode*: ciascun contratto all'interno di un *chaincode* è invocato dalle applicazioni tramite l'identificato unico corrispondente e genera una transazione.

Un concetto peculiare di questo tipo di rete e che lo distingue dalle altre blockchain è quello che Hyperledger Fabric chiama *endorsement policy*. La politica di approvazione è valida per tutti gli *smart contracts* inclusi all'interno di un *chaincode* e definisce quali organizzazioni devono firmare una transazione affinché questa sia considerata valida. Supponiamo che l'*endorsement policy* richieda che una transazione per essere valida debba essere firmata dall'organizzazione *pippo*: se la politica non venisse rispettata, la transazione sarebbe etichettata come non valida. Una transazione di questo tipo non aggiornerebbe i fatti dell'oggetto aziendale nel *world state*, ma verrebbe comunque aggiunta alla blockchain.

Nell'*endorsement policy* vediamo il tentativo più realistico da parte di Hyperledger Fabric di modellare le relazioni aziendali: solo le organizzazioni identificate e ritenute affidabili dalla rete possono convalidare una transazione.

Le politiche per un consorzio di organizzazioni vengono stabilite prima della creazione della rete stessa, ma questo non preclude la possibilità di poterle poi modificare in futuro per adattarle a nuovi requisiti. A questo proposito Hyperledger Fabric definisce il cosiddetto *Fabric chaincode Lifecycle* che si articola in quattro passi:

1. Impacchettamento del *chaincode* da parte di almeno uno delle organizzazioni.
2. Installazione del *chaincode* nel proprio nodo *peer* da parte di ogni organizzazione che intende interrogare o approvare una transazione.
3. Approvazione della definizione del *chaincode* da parte di ogni organizzazione che intende usare il *chaincode*. Di default almeno metà più uno dei partecipanti deve approvare.
4. Se il numero di approvazioni è abbastanza, allora è sufficiente che uno solo dei partecipanti invii la transazione di definizione del *chaincode* sul canale.

Al termine di queste operazioni, gli *smart contracts* all'interno del *chaincode* sono resi disponibili alle applicazioni per essere invocati.

Consenso

In questo paragrafo viene descritto come nella rete di Hyperldger Fabric viene raggiunto il consenso. In particolare, ad alto livello, il flusso di lavoro è noto come *simulate-order-validate-commit* è caratterizzato da quattro fasi [1]:

1. *Simulation phase*. Il processo inizia quando un'applicazione *client* invia una proposta di transazione ai cosiddetti *peers* di *endorsement*. Gli *endorsers* sono presenti in numero almeno pari ad uno per ogni organizzazione coinvolta nel flusso di lavoro. Il compito di questi nodi è simulare autonomamente tramite gli *smart contracts* la proposta di transazione. Questo calcolo viene eseguito rispetto al valore corrente dello stato nel *world state* e genera due vettori ausiliari chiamati *read set* (RS) e *write set* (WS). I due vettori vengono inviati da ciascun nodo di *endorsement* direttamente al *client* che ha fatto la proposta, accompagnati dalla loro firma. Essendo la prima fase di simulazione, i risultati della computazione non apportano alcuna modifica allo stato nel *world state*: in questa fase, infatti, non si ha ancora la certezza che le transazioni generate siano valide. In ogni caso, se tutti i RS e WS ricevuti dal *client* coincidono, quest'ultimo invia il risultato sotto forma di transazione al servizio di ordinamento.
2. *Ordering phase*. Le transazioni ricevute dal servizio di ordinamento vengono disposte all'interno di un blocco secondo un ordine generale unico. La scelta del blocco, anziché delle single transazioni, ha lo scopo di ridurre il carico sulla rete. I nodi di ordinamento non controllano il contenuto delle transazioni, ma hanno il solo compito di organizzarle secondo l'ordine di arrivo. La politica basata sull'ordine di arrivo

ha il vantaggio di essere molto semplice, ma il difetto di creare potenzialmente conflitti e quindi molte transazioni non valide. I blocchi vengono poi inviati a tutti i nodi della rete, in parte direttamente tramite il servizio di ordinamento, in parte tramite il protocollo di *gossip*. Nonostante questa separazione funzionale possa determinare l'arrivo non sequenziale dei blocchi, il servizio garantisce l'esecuzione ordinata.

3. *Validation phase*. Le transazioni contenute nei blocchi vengono convalidate, da ciascun peer, secondo l'*endorsement policy* e i conflitti di serializzabilità. Nel primo caso la verifica utilizza la firma generata dai nodi di *endorsement* durante la fase di simulazione. Nel secondo caso viene valutata la corrispondenza tra i numeri di versione nel RS e nello stato attuale: se questi coincidono significa che lo stato è l'ultimo e quindi la transazione valida.

4. *Commit phase*. Solo in questa ultima fase il blocco viene aggiunto al *ledger* di ciascun *peer* e lo stato del *world state* aggiornato di conseguenza.

L'intero processo risulta essenziale per garantire che l'ordine delle transazioni sia rispettato da tutti i nodi della rete.

3.3 Scelte implementative

Versione

L'applicazione è stata sviluppata utilizzando la versione v2.2.4 di Hyperledger Fabric rilasciata l'8 settembre 2021. La motivazione è che la versione v2.2 rilasciata il 20 luglio 2020 è la prima a essere supportata per un lungo periodo (LTS). Le versione successive v2.2.x hanno come scopo la correzioni di eventuali errori.

Architettura

Osserviamo la rete in figura. La rete N viene avviata da B che configura e ospita il servizio di ordinamento OB. I diritti amministrativi sulla rete N sono assegnati a B tramite la configurazione di rete NCB che lo identifica come amministratore unico della rete. L'autorità di certificazione CAB assegna certificati X.509 alle entità che compongono l'organizzazione B.

B, in qualità di amministratore unico della rete, è il solo che può creare consorzi. I consorzi sono dei sottoinsiemi di organizzazioni che intendono effettuare tra loro delle negoziazioni e che vengono raggruppati per questo scopo. A questo punto B crea un consorzio composta da due organizzazioni: Org1 e Org2. L'organizzazione Org1 è costituita dagli enti di certificazione che rilasciano certificati, mentre l'Org2 è composta dai clienti che intendono certificarsi. B non è stato incluso nel consorzio in quanto non partecipa a questo tipo di negoziazioni. In altre parole possiamo considerare i membri dell'Org1 come i clienti di B, mentre i membri dell'Org2 come i clienti dell'Org1. Facendo riferimento all'architettura sarebbe lecito considerare anche i membri dell'Org2 come clienti di B, ma questo non sarebbe del tutto vero dal punto di vista funzionale: come sarà più chiaro

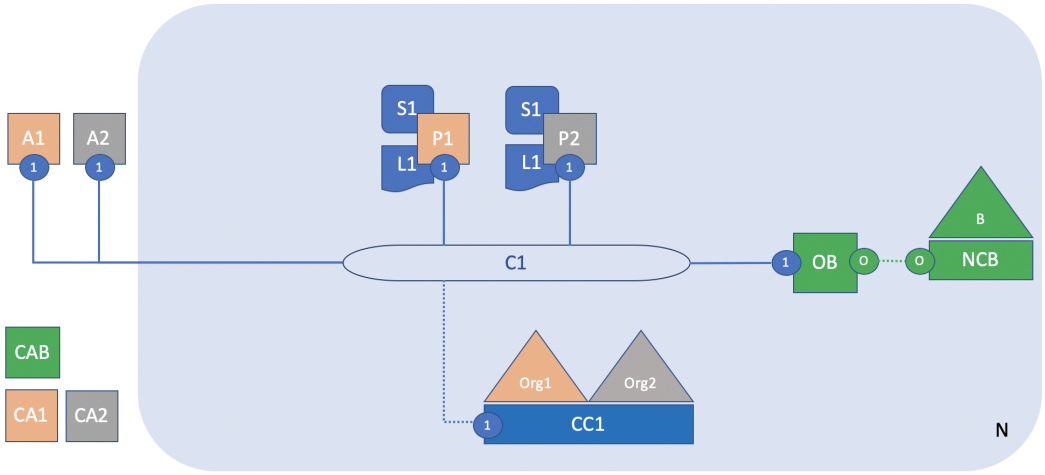

Figura 3.2. La rete N è composta da tre organizzazioni: B, Org1 e Org2. B è Bechain, Org1 è l'insieme degli enti di certificazione e Org2 è l'organizzazione dei clienti. B ospita il nodo di ordinamento OB. Org1 e Org2 comunicano privatamente tramite i rispettivi peer P1 e P2 connessi tramite il canale privato C1. Ciascun peer possiede il ledger L1 e il chaincode S1 invocati rispettivamente tramite le applicazioni client A1 e A2. B è sottoposto alla configurazione di rete NCB, Org1 e Org2 agiscono secondo la politica di canale CC1. Infine, ciascuna organizzazione ha la propria Certification Authority CA.

in seguito, l'inserimento della blockchain nel contesto delle certificazioni ha come scopo quello di migliorare lo strumento di accreditamento senza compromettere l'esperienza per i clienti. Parallelamente alla definizione del consorzio, per ciascuna organizzazione viene individuata un'autorità di certificazione. Anche in questo caso lo scopo di CA1 e CA2 è identificare i componenti delle rispettive organizzazioni.

Successivamente Org1 e Org2 per poter comunicare privatamente all'interno della rete N creano un canale C1. Il canale garantisce, all'interno della stessa infrastruttura condivisa, riservatezza all'interno della rete e rispetto ad altri canali. Il canale C1 obbedisce alla configurazione di canale CC1 che è completamente disgiunta dalla configurazione NCB. CC1 definisce i diritti che l'Org1 e l'Org2 possiedono nei confronti del canale C1. A questo proposito notiamo che B pur essendo amministratore della rete, non fa parte del consorzio e quindi non ha privilegi a livello di canale CC1 che è al contrario dipende dalle due organizzazioni. Pertanto alcune funzionalità come l'aggiunta di nuove organizzazioni al consorzio o la modifica della configurazione del canale non potranno essere gestite da B. Analogamente qualsiasi cambiamento alla configurazione NCB da parte di B non ha ripercussioni sulla configurazione CC1. La scelta di escludere Bechain è mossa dall'esigenza di mantenere confidenziali le transazioni tra gli enti di certificazione e i rispettivi clienti. Infatti, il ruolo di B si limita a quello di fornire l'infrastruttura di rete e non avrebbe senso includerlo come terza parte fidata.

La fase successiva prevede l'aggiunta di un nodo *peer* per ciascuna delle due organizzazioni che compongono il consorzio. Precisamente i nodi sono P1 e P2 ed entrambi sono composti dal *chaincode* S1 e da una istanza del *ledger* L1. Parliamo d'istanza perché

il *ledger* è distribuito sui nodi *peer* di ciascuna organizzazione, ma è logicamente unico per l'intero canale C1. Ciò che permette di identificare i *peesr* delle due organizzazioni è ancora una volta il certificato digitale emesso dalle rispettive CA1 e CA2. Durante la fase di ordinamento, OB utilizza la configurazione CC1 per identificare se i *peers* P1 e P2 possono effettuare letture o scritture sul canale. A proposito dei nodi *peer* è necessario fare un'osservazione: in questa rete sono stati introdotti unicamente due nodi *peer* ed entrambi ospitano un *chaincode*. In reti di grandi dimensioni il numero di *peers* potrebbe essere aumentato per incrementare le prestazioni e per garantire una maggiore resilienza in caso di guasti o interruzioni programmate. In questo modo diverse applicazioni potranno connettersi a *peer* diversi. Inoltre non è necessario che tutti i *peers* ospitino uno *smart contract*: chi non lo possiede può comunque partecipare alla fase di convalida e il protocollo di *gossip* garantisce una comunicazione efficiente.

L'ultima fase prevede lo sviluppo di un'applicazione web che permette alle due organizzazioni di connettersi ciascuno al proprio *peer*. Questa situazioni è rappresentata in figura introducendo due applicazioni *client* chiamate rispettivamente A1 e A2. Anche in questo caso l'appartenenza di ciascun componente alla propria organizzazione è garantita dal certificato X.509 emesso dalla rispettiva CA. A questo punto, ciascuna applicazione può invocare lo *smart contract* del *chaincode* S1, definito sul proprio *peer*, per interrogare o aggiornare il registro L1. Notiamo che, rispetto ai casi precedenti, in questa circostanza l'applicazione web è esterna alla rete Fabric, ma ciò non costituisce un problema dal momento che l'accesso alla rete è garantito dal canale. Per concludere, invero, osserviamo che il canale è il punto di connessione per tutti i componenti fino a qui introdotti, vale a dire nodi *peers*, nodi *orderers* e *applicazioni client*.

È necessario aggiungere un commento sul servizio di ordinamento OB ospitato da B. OB non possiede un *ledger*, ma sappiamo che una delle sue funzioni è riordinare le transazioni provenienti dai nodi di *endorsement*. In questo caso i nodi *peer* sono i nodi P1 e P2 delle due organizzazioni e la privacy non è perciò del tutto garantita. Dobbiamo però fare una considerazione: nella fase di ordinamento le transazioni non sono ancora convalidate pertanto B non sa se sono corrette. Questo, purtroppo, non risolve il problema completamente, pertanto nei paragrafi successivi verrà esposta una soluzione basata sulla crittografia.

Identità

Per creare nella rete blockchain la CAB, la CA1 e la CA2 si utilizza un competente integrato di Hyperledger Fabric. Questo componente si chiama Fabric CA ed è una CA radice privata che assegna attraverso certificati standard X.509 l'identità ai diversi componenti della rete. Ad ogni identità può essere assegnato uno dei seguenti quattro ruoli:

- *Member*. Indica un membro qualunque di una specifica organizzazione.
- Peer. In questo caso il nodo può essere di *endorsement* o di *commit*.
- *Admin*. Assegna all'utente privilegi di amministratore.

- *Client*. Permette l'invocazione dei contratti intelligenti nella blockchain. Durante la fase di registrazione degli enti di certificazione e dei clienti si dovrà assegnare il ruolo di *Client* per consentire l'esecuzione delle transazione.

Database di stato

Hyperledger Fabric offre due possibili alternative per l'implementazione dello stato mondiale. In particolare, è possibile scegliere tra due database di stato: LevelDB e CouchDB. LevelDB permette di rappresentare gli stati come coppie chiave-valore e supporta un set limitato di operazioni. Le principali operazioni sono la PUT, la DELETE e la GET. Nel caso delle interrogazioni, è possibile effettuare *query* basate sulla chiave, su un range di chiavi o su una chiave composta, ma non sono supportate *query* SQL. I principali vantaggi nell'usare un database di questo tipo sono la semplicità e la velocità.

CouchDB, al contrario, memorizza i dati come documenti nel formato json e permette una grande varietà d'interrogazioni basate sui valori invece che sulle chiavi. Inoltre, supporta indici e paginazione e i dati sono resi disponibili tramite HTTP URI. Per questa ragione è possibile usare le operazioni fornite da HTTP come GET, PUT, POST ecc. Il suo utilizzo è particolarmente consigliato quando dobbiamo interrogare grandi set di dati.

Per questo progetto si è deciso di usare come database di stato la soluzione proposta di default da Hyperledger Fabric. L'idea di utilizzare LevelDB è mossa dalle seguenti motivazioni: non essendo l'insieme dei dati particolarmente esteso, non vi è la necessità di usare CouchDB come database di stato. L'utilizzo del database LevelDB e della chiave composta soddisfa tutti i requisiti richiesti dall'applicazione. L'utilizzo di CouchDB risulterebbe preferibile nel momento in cui la complessità dell'applicazione e il flusso di transazioni aumentasse. Bisogna infatti specificare che CouchDB garantisce le stesse funzionalità del database LevelDB. Inoltre, in un ambiente di produzione, l'utilizzo di un modello astratto quale json favorirebbe la verifica formale e l'*audit*.

Chiave composta

All'interno del *ledger*, il *world state* è composto da una o più coppie chiave-valore. In questo paragrafo analizzeremo la composizione della chiave motivando l'implementazione.

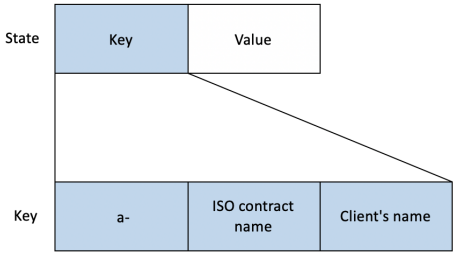

Figura 3.3. Chiave composta.

Hyperledger Fabric permette di creare la cosiddetta *composite key*. Per prima cosa concentriamoci sulla sua composizione e notiamo che la chiave stessa è suddivisa in tre parti:

1. Il suffisso a-.
2. Il nome del contratto.
3. Il nome del cliente.

La scelta di questi campi comporta un primo risultato nell'ambito della sicurezza. La riservatezza delle informazioni di ciascun cliente è garantita attraverso il confronto tra la terza parte della chiave e l'identità contenuta all'interno del certificato X.509. Se la corrispondenza con l'identità non è verificata, significa che l'utente sta cercando d'interrogare o aggiornare un *asset* che non è di sua proprietà. Al verificarsi di questa circostanza lo *smart contract* impedisce immediatamente l'esecuzione.

Il secondo risultato riguarda l'ambito delle interrogazioni. Nel *world state* ogni stato è costituita da una terna univoca, mentre nel caso di chiavi parziali il numero di corrispondenze è pari o maggiore a uno. Il motivo per cui la chiave è stata scomposta in parti è per effettuare *query* più efficienti nel database di stato. In caso contrario, infatti, sarebbe stato necessario leggere tutti i record e poi effettuare un'analisi fuori catena. Osserviamo queste conclusioni nelle due immagini seguenti:

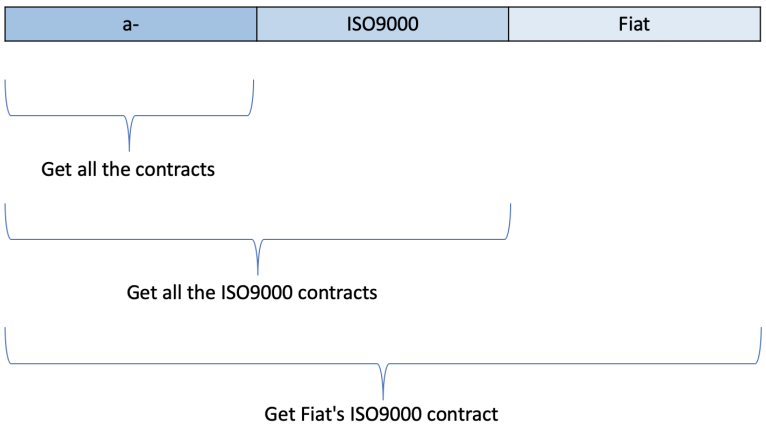

Figura 3.4. Interrogazione della base di dati lato istituzione.

Supponiamo di essere un'istituzione e di voler effettuare una ricerca nel *world state* sulla base della chiave raffigurata: la chiave è composta dal suffiso -a, dal nome del contratto ISO9001 e dal nome del cliente Fiat. Immaginiamo che lo stato mondiale sia composto da un certo numero di stati ognuno con la propria chiave e valore: la suddivisione della chiave in parti permette di effettuare interrogazioni a tre livelli di granularità:

1. Includendo sola la prima parte della chiave, cioè il suffisso "-a", è possibile interrogare tutti gli stati presenti nello stato mondiale.

2. Includendo il suffisso e il nome del contratto è possibile effettuare una ricerca di tutti i contratti ISO9001 presenti nello stato mondiale. Si noti che essendo l'interrogazione effettuata da un ente di certificazione è probabile che nel *world state* siano presenti più processi di certificazione in parallelo per uno stesso contratto, ma relativi a clienti diversi.

3. Includendo tutte le parti della chiave, quindi aggiungendo anche il nome del cliente Fiat, è possibile effettuare una ricerca basata su una chiave specifica.

Risulta evidente l'importanza di avere una chiave composta: nel caso di un'istituzione ci saranno molti processi di certificazione in parallelo e la possibilità di filtrare l'output è molto utile.

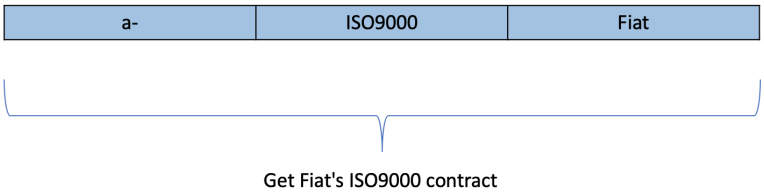

Figura 3.5. Interrogazione della base di dati lato cliente.

Nell'immagine seguente analizziamo la stessa interrogazione, ma dal punto di vista di un cliente. Nonostante sia possibile, anche in questo caso, effettuare ricerche a diversi livelli di granularità, il risultato non sarebbe soddisfacente. In questo caso è ragionevole supporre che il cliente Fiat avvii processi di certificazione unici per certificati diversi. Pertanto, risultano utili solo due tipi di ricerche: la prima riguarda uno specifico stato nel *world state*, la seconda riguarda l'elenco di tutti gli stati attualmente attivi.

Struttura dati

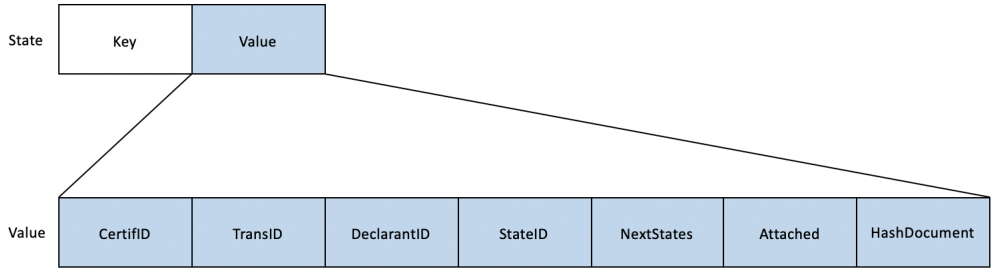

Figura 3.6. Valori associati a una chiave composta.

Passiamo ora ad analizzare il valore associato a ciascuna chiave. Il valore è composto da diversi campi e ogni campo include informazioni di carattere generale che possono essere adattarsi a qualsiasi tipologia di certificazione. Di seguito un'analisi più dettagliata:

- CertifID: identificatore univoco o chiave composta.
- TransID: id della transazione.
- DeclarantID: identificativo o nome dell'attore che ha effettuato la transazione.
- StateID: stato attuale della macchina a stati.
- NextState: prossimo stato della macchina a stati raggiungibile a partire dallo stato attuale.
- Attached: valore booleano che indica se un allegato è richiesto o meno.
- HashDocument: hash del documento inviato come allegato più il sale.

La scelta d'inserire l'hash del documento più il sale è motivata da diverse riflessioni. Le prestazioni di un sistema basato sulla blockchain dipendono da diversi fattori tra cui la dimensione delle transazioni e dei blocchi. Esiste uno studio a riguardo [4]. Per questa ragione si è preferito includere l'hash del documento piuttosto che il documento stesso. Inoltre, ricordando che il servizio di ordinamento è ospitato da Bechain, nei paragrafi precedenti si era sollevato il problema riguardo cui Bechain potesse leggere le transazioni non ancora convalidate. L'idea d'includere l'hash del documento ha come obiettivo impedire a Bechain di comprendere le transazioni. Mentre l'idea d'includere anche il sale ha come scopo quello di ostacolare la possibilità di risalire al documento tramite l'hash. Infatti, se più di un documento fosse compilato allo stesso modo, l'hash risultante sarebbe il medesimo. A questo punto risultano necessari due ulteriori passaggi: da un lato è richiesto che ogni istituzione memorizzi i documenti nel proprio database locale, dall'altro è necessario che il riferimento a questi documenti sia garantito tramite una chiave il cui valore coincide con il valore della chiave usata nella blockchain. In questo modo, nel caso in cui il documento originale fosse rimosso dal database, sarebbe possibile risalire, tramite la blockchain, all'hash del documento, come prova inconfutabile della sua presenza.

Chaincode

In questo paragrafo viene descritta l'implementazione del *chaincode*.

Per prima cosa consideriamo il canale introdotto nel paragrafo riguardante l'architettura. Questo componente permette una comunicazione privata tra i membri che lo costituiscono e, di conseguenza, limita la visibilità ai soli membri dell'Org1 e dell'Org2. Questa implementazione permette di escludere Bechain dalle negoziazioni e di mantenere riservate le informazioni scambiate tra gli enti di certificazione e i propri clienti. La scelta è legittimata dal fatto che da un lato Bechain non partecipa alle transazioni aziendali, dall'altro la sua inclusione come terza parte fidata non garantisce il suo corretto comportamento. Ad esempio Bechain potrebbe rilevare i dati sensibili o i dettagli di una stipulazione contrattuale.

Consideriamo ora un altro aspetto e analizziamo con maggior dettaglio le organizzazioni Org1 e Org2. Fino a questo momento abbiamo descritto l'Org1 come l'insieme delle

istituzioni che distribuiscono certificati e l'Org2 come l'insieme dei clienti che richiedono una certificazione. Un'architettura di questo tipo fornisce visibilità a tutti i membri delle organizzazioni riguardo ogni tipo di negoziazione effettuata. Ad esempio, supponiamo che l'*ist1* stia scambiando informazioni sia con il *client1* che con il *client2* e che, parallelamente, l'*ist2* stia scambiando informazioni con il *client3*. Il modello fino a ora descritto permetterebbe, ad esempio, al *client2* di accedere ai dati scambiati tra l'*ist1* e il *client1* e all'*ist2* di accedere ai contenuti riservati scambiati dall'*ist1* con ognuno dei suoi clienti. Risulta evidente che questa situazione è inaccettabile. Per gestire questa problematica sono state prese in considerazione due possibili soluzioni. Nel primo caso si è valutata la possibilità di creare un canale di comunicazione privato tra ogni istituzione e cliente: il canale per definizione assicura una comunicazione privata e quindi certamente risolve il problema. Questa ipotesi è stata successivamente scartata in quanto la creazione di canali multipli avrebbe comportato un sovraccarico dovuto alla creazione e al mantenimento di molti canali separati. La seconda soluzione ha preso in considerazione l'utilizzo dei cosiddetti *chaincode namespaces*. Lo spazio dei nomi *chaincode* garantisce un'importanti proprietà per cui ogni *chaincode* definito nel canale è caratterizzato da un proprio *world state*. Questo significa che tutti gli *smart contracts* definiti all'interno di uno stesso *chaincode* possono accedere solamente allo stato mondiale riservato al *chaincode* di appartenenza. Lo spazio dei nomi permette quindi di separare lo stato mondiale del *ledger* e impedire che gli *smart contracts* di un *chaincode* possano accedere al *world state* di altri *chaincode*.

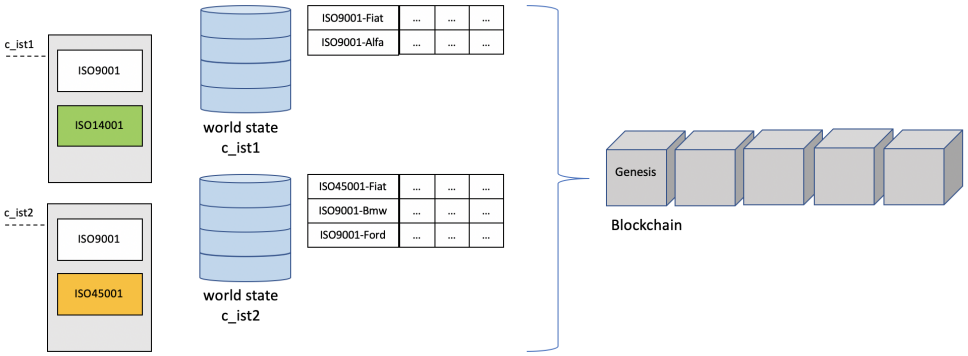

Figura 3.7. *Chaincode namespaces.*

Per capire in che modo la separazione dei nomi ci può venire in aiuto, osserviamo la rappresentazione grafica. L'Org1 è l'organizzazione che include tutti gli enti di certificazione. Per semplicità consideriamo solo due di queste istituzioni e le chiamiamo *ist1* e *ist2*. In questo esempio l'*ist1* rilascia due tipologie di certificati, l'ISO 9001 e l'ISO 14001; mentre l'*ist2* oltre all'ISO 9001 rilascia anche la certificazione ISO 45001. In entrambi i casi i processi di certificazione relativi a questi contratti sono modellati da dei contratti intelligenti impacchettati all'interno dei *chaincodes* di appartenenza. Nello specifico *c-ist1* rappresenta il *chaincode* dell'*ist1*, mentre *c-ist2* rappresenta il *chaincode* dell'*ist2*. Come possiamo vedere in figura, ciascun *chaincode* possiede il proprio *world state* separato e,

di conseguenza, ciascun ente di certificazione ha visibilità solamente sul proprio dominio. Con questa soluzione abbiamo ottenuto un importante risultato cioè che ogni ente di certificazione mantiene confidenziali le proprie informazioni.

Nonostante la separazione dello stato mondiale del *ledger*, tutte le transazioni effettuate dai membri di Org1 e Org2 sono memorizzate all'interno della stessa struttura blockchain. Infatti, la blockchain è logicamente unica per il canale e quindi contiene le transazioni relative a diversi *world states*. Con questa precisazione è lecito obiettare che l'accesso alla blockchain da parte dei membri possa violare il vincolo di riservatezza. In realtà questa considerazione non è corretta in quanto ogni utente ha una visibilità limitata ai soli *assets* del proprio stato mondiale e quindi non può in alcun modo interrogare elementi al di fuori dal proprio dominio. Inoltre, anche il vincolo di *ownership* degli *assets* non lo consentirebbe: a questo proposito si faccia riferimento al paragrafo riguardante la chiave composta.

Smart Contracts

Dopo aver discusso nel capitolo precedente l'implementazione dei *chaincodes*, definiamo ora gli *smart contracts*. Ogni istituzione possiede un proprio *chaincode* e ogni *chaincode* è costituito da uno o più contratti intelligenti. La logica degli *smart contracts* definisce la macchina a stati dei processi di certificazione e fornisce funzionalità aggiuntive che permettono d'interrogare lo stato mondiale e la blockchain e annullare logicamente una transazione. In questo ultimo caso si distinguono gli errori umani dalle violazioni contrattuali e l'annullamento è logico in quanto le transazioni fisiche sulla blockchain sono permanenti.

Politiche

La terza fase del *Fabric chaincode Lifecycle* prevede l'approvazione della definizione del *chaincode*. Hyperledger Fabric definisce la cosiddetta *LifecycleEndorsement policy* per definire quali organizzazioni devono partecipare al processo di approvazione. L'impostazione prevista di default da Hyperledger Fabric, utilizza come *Type* il valore *ImplicitMeta* e come *Rule* il valore *MAJORITY Endorsement*. Essendo presenti sul canale due organizzazioni, la politica di default si adatta correttamente. Infatti, la maggioranza di due è due e quindi sia Org1 che Org2 devono approvare il *chaincode* per poter usare gli *smart contracts* sul canale.

Consideriamo ora l'*endorsement policy* e ricordiamo che ogni contratto intelligente obbedisce all'*endorsement policy* definita per il *chaincode* di appartenenza. La politica di approvazione indica quante organizzazioni devono accettare con i propri *peers* di *endorsement* una transazione affinché quest'ultima possa essere convalidata. Anche in questo caso l'utilizzo d'*ImplicitMeta* e *MAJORITY Endorsement* è giustificato perché richiede a Org1 e a Org2 di approvare entrambi le transazioni.

Se volessimo includere per ridondanza più *endorsers* all'interno di ciascuna organizzazione sarebbe necessario modificare la *policy* per includere anche questi *peers*. Ad esempio *OutOf(2, 'Org1.peer', 'Org2.peer')* indica che due peers all'interno dell'Org1 e un peer all'interno dell'Org2 devo partecipare al processo di approvazione.

Applicazione web

Per l'implementazione delle applicazioni *client* A1 e A2 si è scelto di realizzare un'applicazione web. L'interfaccia grafica ricorda quella di un'applicazione *blockchain explorer*, ma con l'aggiunta di funzionalità specifiche per il programma. In particolare, l'applicativo consente di:

- registrarsi alla rete blockchain;
- invitare un nuovo cliente;
- avviare un processo di certificazione;
- effettuare una transazione;
- annullare l'ultima transazione;
- visualizzare le transazioni precedenti;
- effettuare una ricerca sulla base del cliente;
- effettuare una ricerca sulla base del certificato;
- effettuare una ricerca sulla base di chiavi multiple.

Capitolo 4

Strumenti di supporto

4.1 Requisiti

Nel paragrafo riguardante il modello astratto è stata introdotta la macchina a stati relativa a un processo di certificazione generico. L'obiettivo è stato quello d'individuare un modello unico che potesse adattarsi alle diverse tipologie di certificazione e alla loro evoluzione nel tempo. Nonostante questi propositi, alcuni processi di certificazione potrebbero differire dal modello unico per uno o più *step* aggiuntivi.

4.2 Funzionalità

Per rispondere a queste esigenze e nella volontà di lasciare libero arbitrio a Bechain nell'implementazione della macchina a stati, sono stati creati due strumenti di supporto. Il primo è chiamato *createGraphviz* e genera un file che è sintatticamente compatibile con l'input richiesto dal software Graphviz [9]. Tramite questo software è stato possibile rappresentare le informazioni strutturate come grafi e ottenere la macchina a stati desiderata. Successivamente questi dati sono stati utilizzati come punto di partenza del secondo applicativo. Quest'ultimo si chiama *createContracts* e permette, con un solo comando, di creare il codice relativo a un intero chaincode.

La validità di questi programmi si esprime anche nella loro capacità di fornire agli amministratori degli strumenti in grado di adattare l'applicazione alle proprie esigenze di *business*. Inoltre, non è richiesta alcuna capacità di programmazione o di sviluppo di *smart contracts* in generale.

4.3 Implementazione

4.3.1 Creazione di una macchina a stati

In questo paragrafo viene mostrato, passo dopo passo, il funzionamento dello script *createGraphiz*.

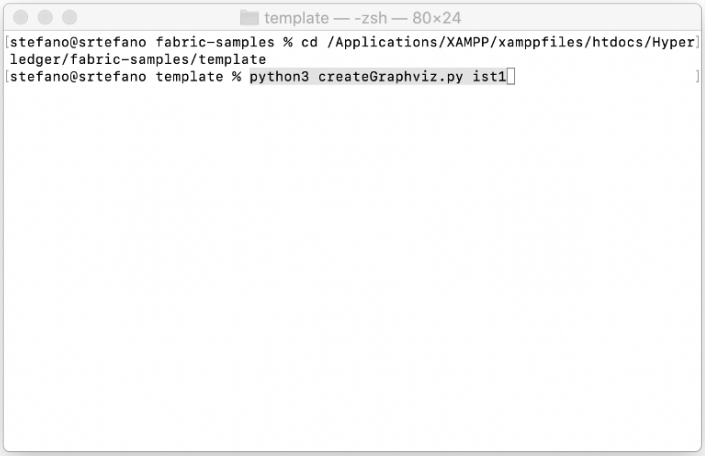

Figura 4.1. Eseguiamo lo script python *createGraphviz.py*.

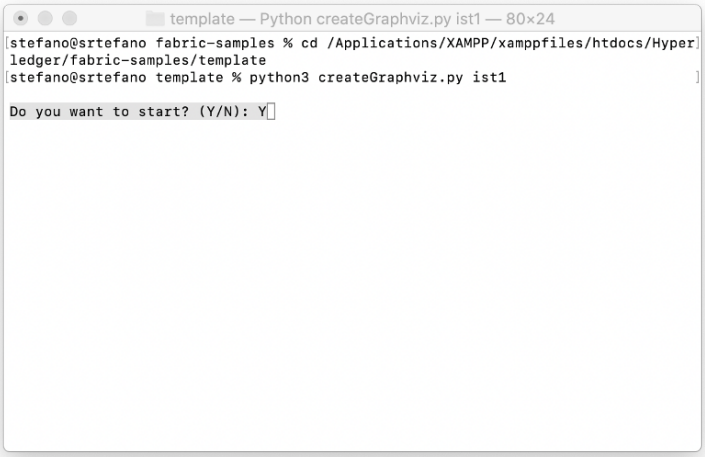

Figura 4.2. Digitiamo Y, cioè *yes*, per avviare lo script.

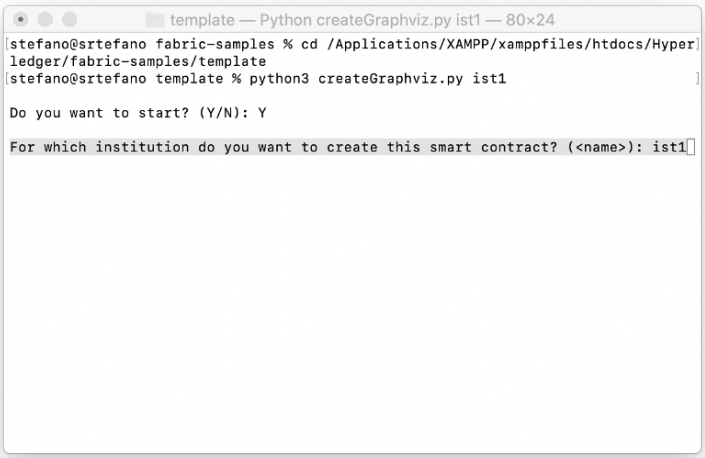

Figura 4.3. Digitiamo il nome dell'istituzione per la quale stiamo definendo la macchina a stati, in questo caso *ist1*.

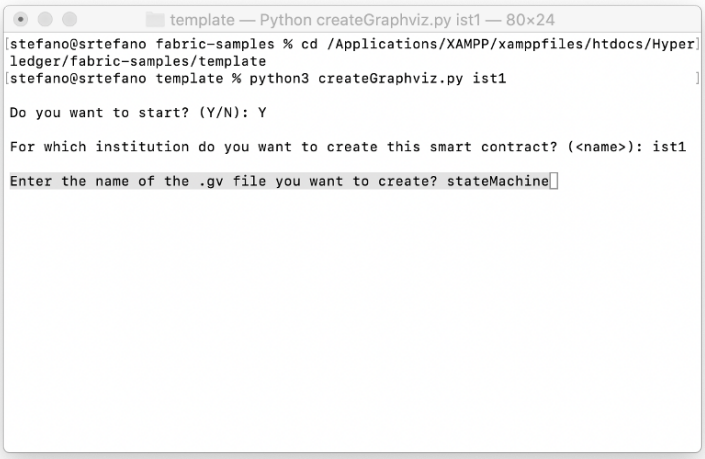

Figura 4.4. Digitiamo il nome che verrà assegnato al file *.gv*, in questo caso *stateMachine*.

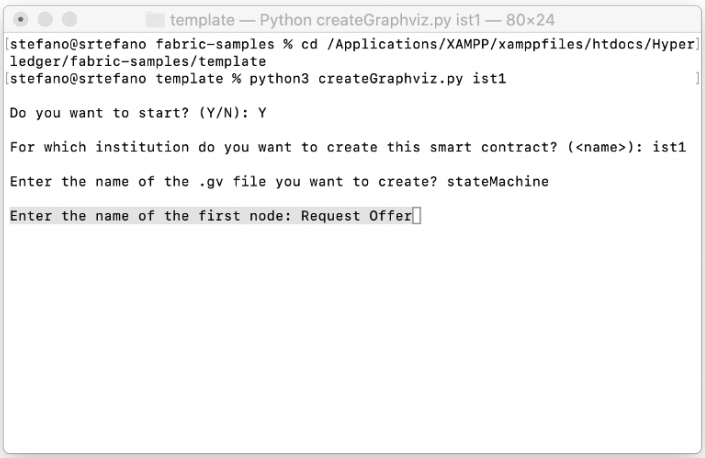

Figura 4.5. Digitiamo il nome del nodo radice, in questo caso *Request Offer*.

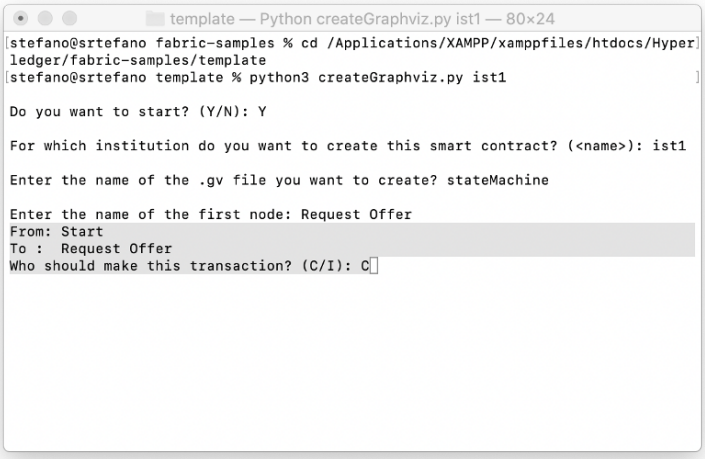

Figura 4.6. Digitiamo il nome di chi dovrà effettuare la transazione di *Request Offer*. La scelta è tra cliente e istituzione. In questo caso digitiamo la *C* di cliente.

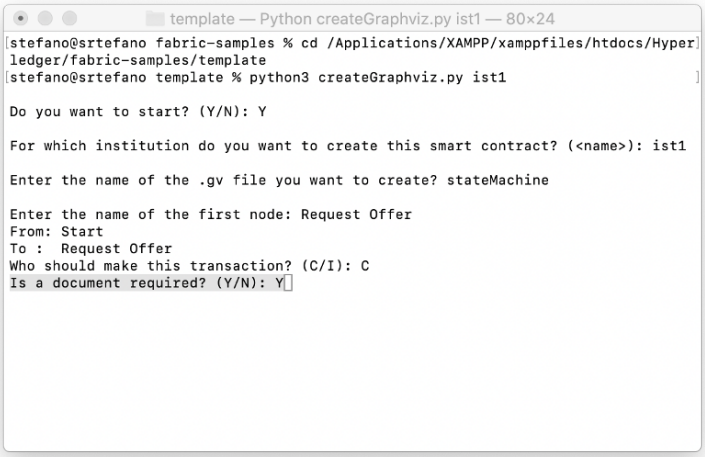

Figura 4.7. Indichiamo se è richiesto un documento allegato alla transazione, in questo caso digitiamo Y.

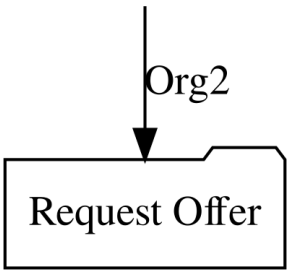

Figura 4.8. Rappresentazione grafica delle scelte precedenti. *Org2* significa che questa transazione deve essere eseguita dal cliente (questa scritta non verrà mostrata all'utente finale). La forma a cartella del nodo indica che è necessario fornire un documento allegato.

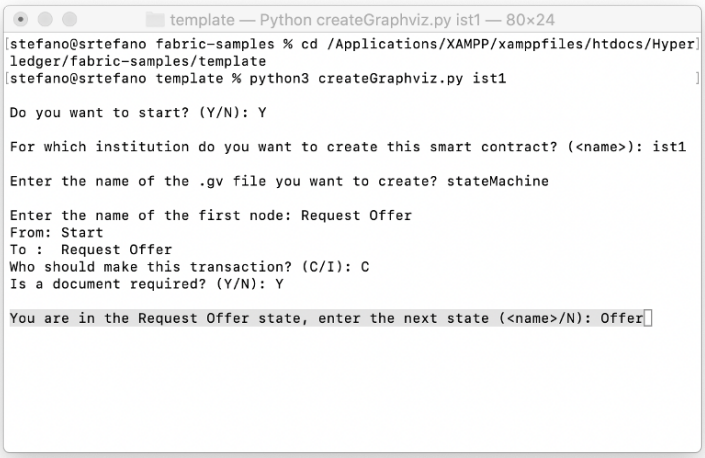

Figura 4.9. Digitiamo *Offer* che sarà il nodo figlio del nodo radice *Request Offer*.

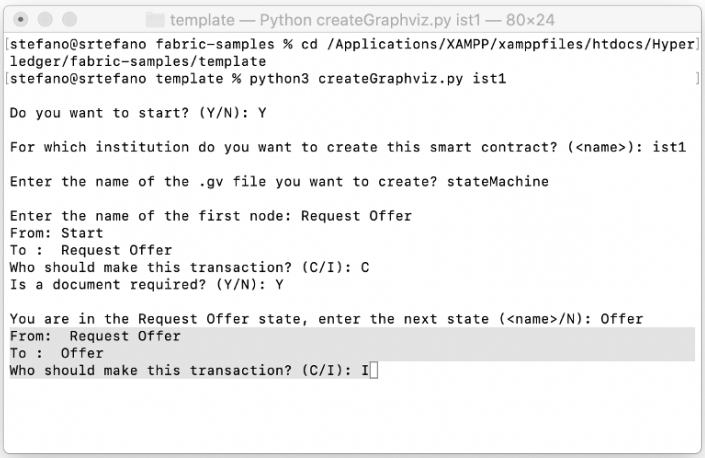

Figura 4.10. In questo caso è l'istituzione che effettuerà l'offerta al cliente, quindi digitiamo *I*.

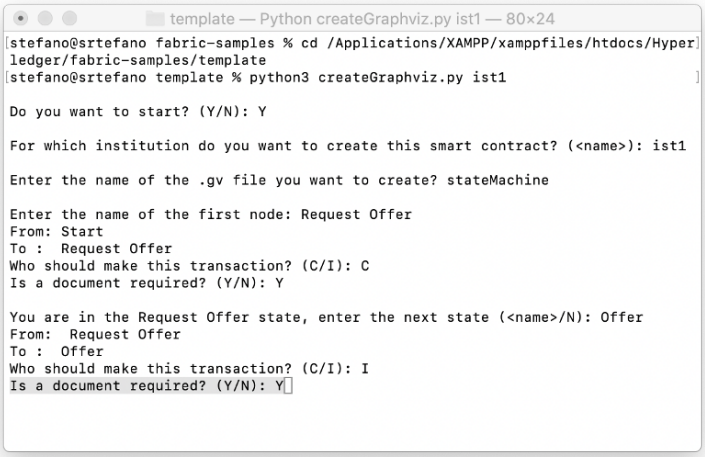

Figura 4.11. La transazione di *Offer* richiede un documento allegato, digitiamo *Y*.

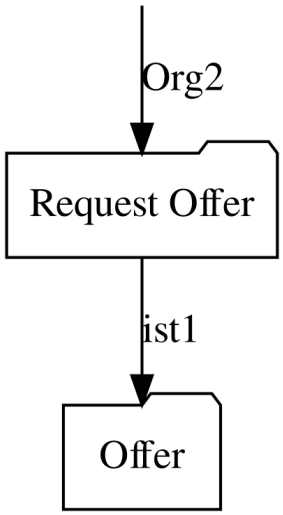

Figura 4.12. Visualizzazione grafica delle scelte precedenti. Il nome dell'arco *ist1* indica che la transazione di offerta deve essere eseguita dall'istituzione. La forma a cartella del nodo indica che è necessario allegare un documento.

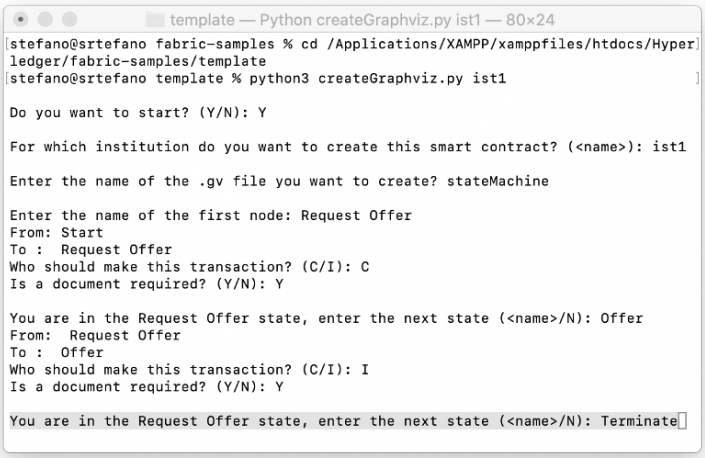

Figura 4.13. Aggiungiamo il nodo foglia *Terminate*.

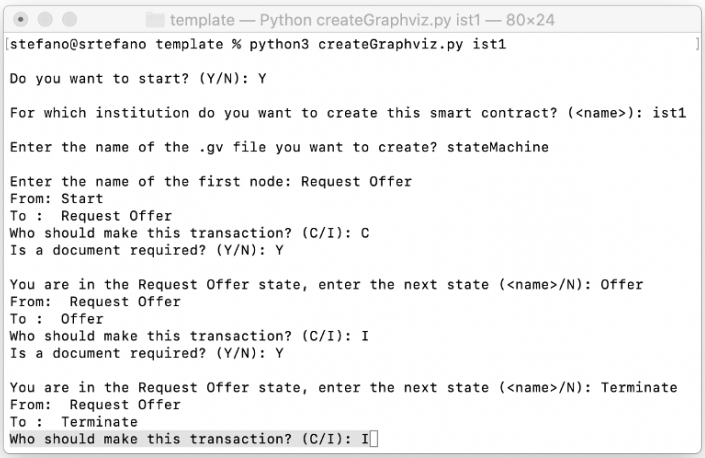

Figura 4.14. La transazione è effettuata dall'istituzione.

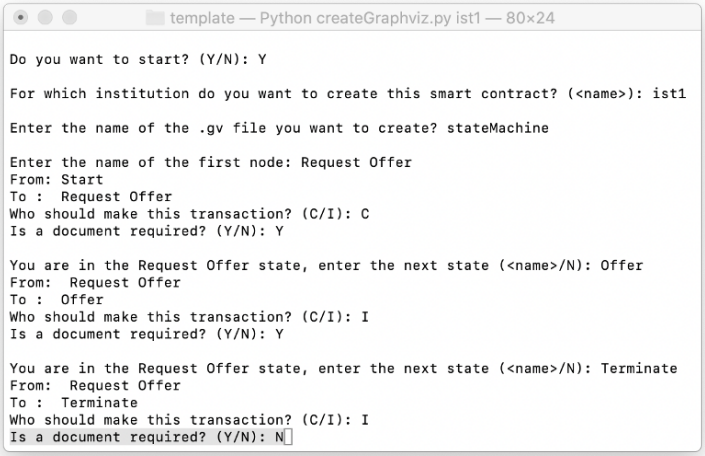

Figura 4.15. Non è richiesto un documento.

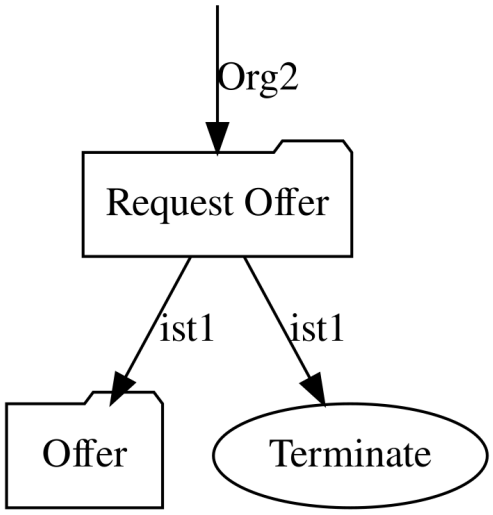

Figura 4.16. Rappresentazione grafica delle scelte precedenti. Non essendo richiesto un documento, il nodo *Terminate* ha la forma di un'ellisse.

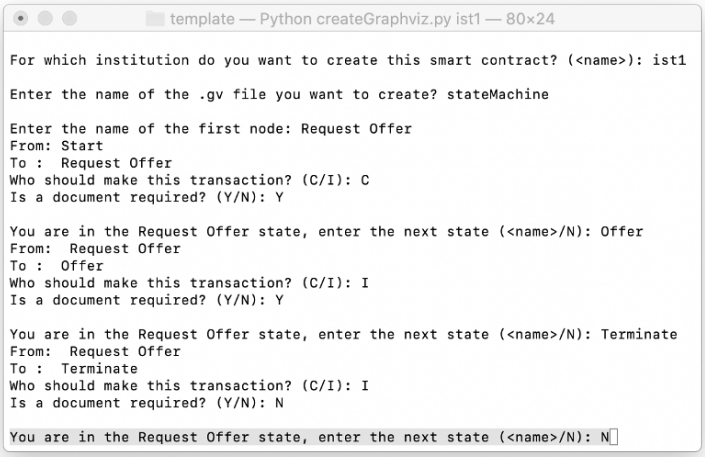

Figura 4.17. Se il nodo radice non ha altri nodi foglia, digitiamo *N* e proseguiamo la visita in profondità del grafo con il prossimo nodo.

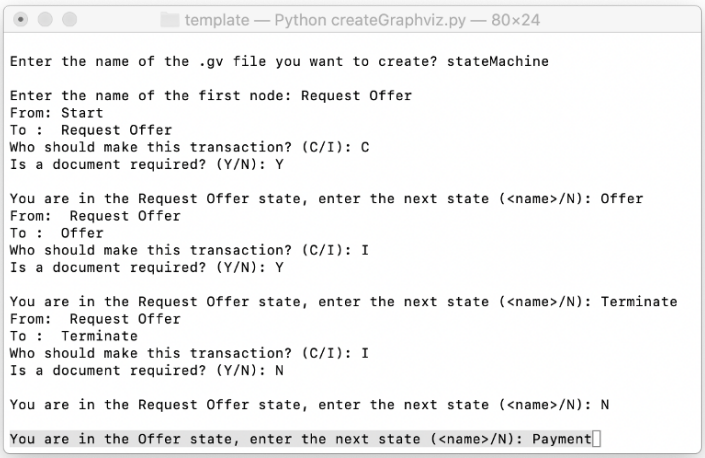

Figura 4.18. Continuiamo eseguendo le stesse operazioni per ciascun nodo foglia. Partiamo dal nodo *Offer* e digitiamo *Payment*.

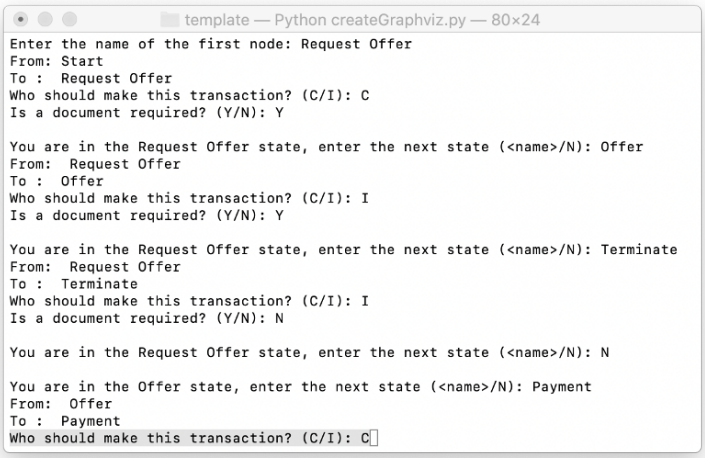

Figura 4.19. La transazione è effettuata dal cliente.

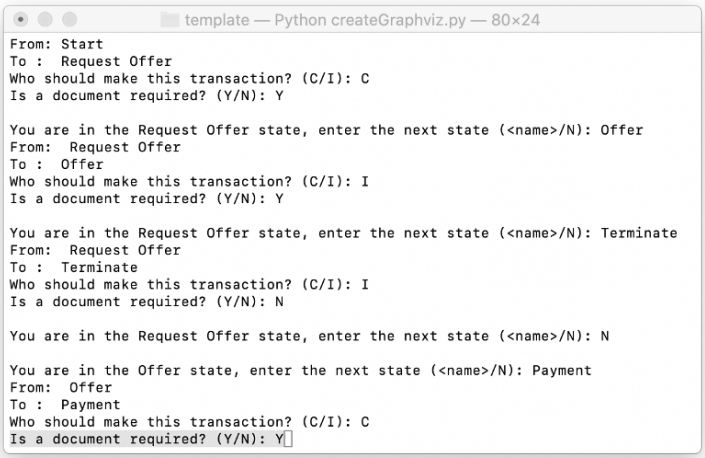

Figura 4.20. La transazione richiede un documento allegato.

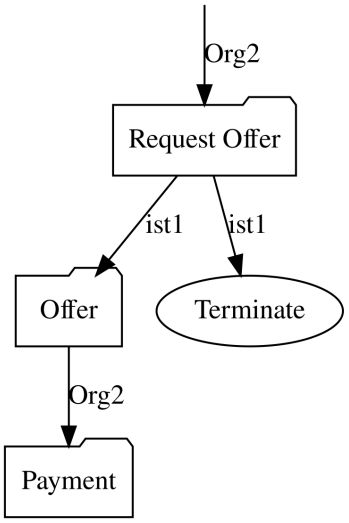

Figura 4.21. Rappresentazione grafica delle scelte precedenti.

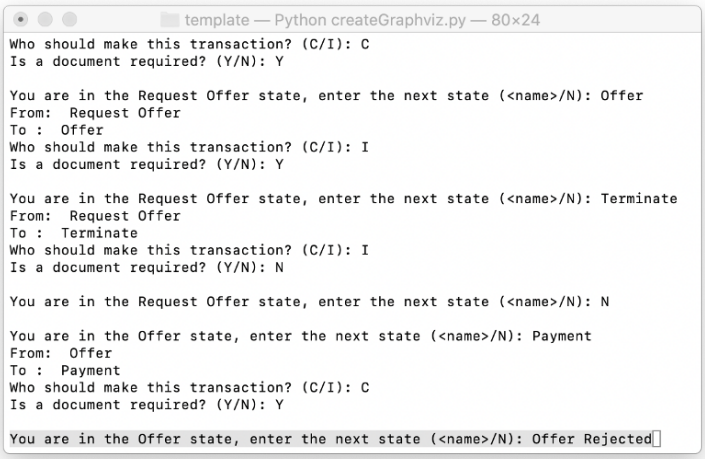

Figura 4.22. Aggiungiamo il nodo *Offer Rejected*.

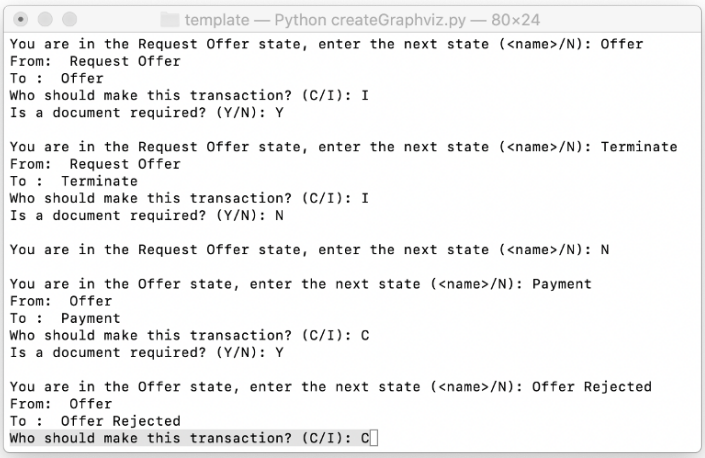

Figura 4.23. La transazione è effettuata dal cliente.

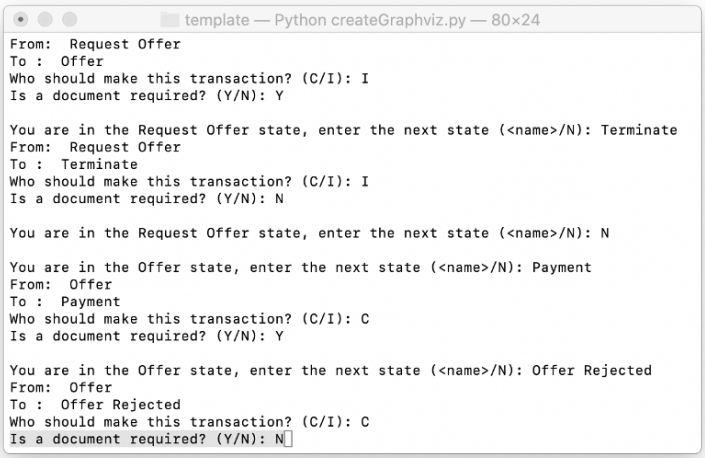

Figura 4.24. La transazione non richiede un documento allegato.

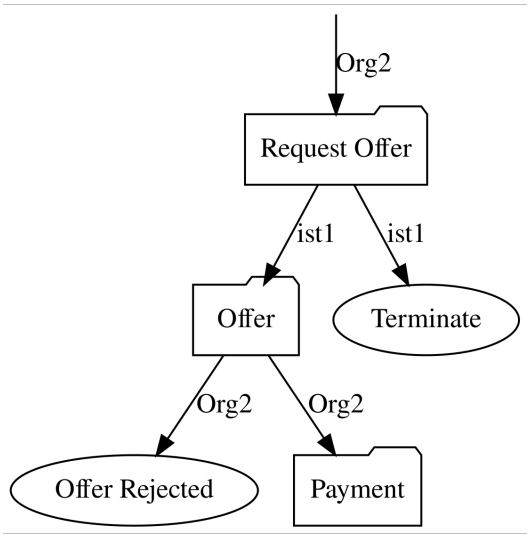

Figura 4.25. Rappresentazione grafica delle scelte precedenti.

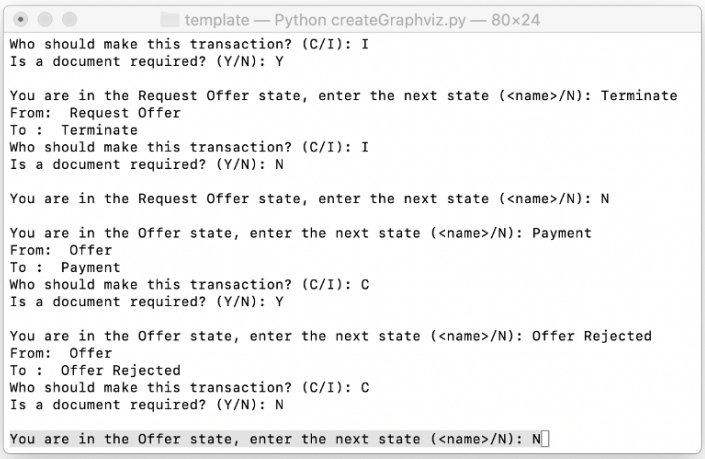

Figura 4.26. Non sono presenti ulteriori nodi foglia.

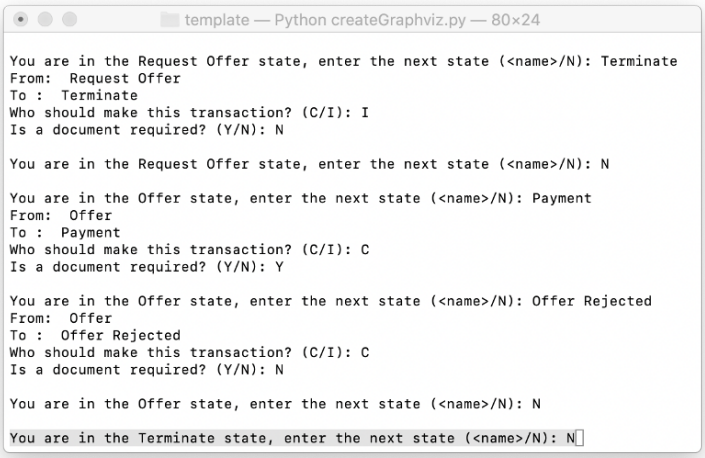

Figura 4.27. Siamo nel nodo *Terminate*, non sono presenti ulteriori nodi foglia.

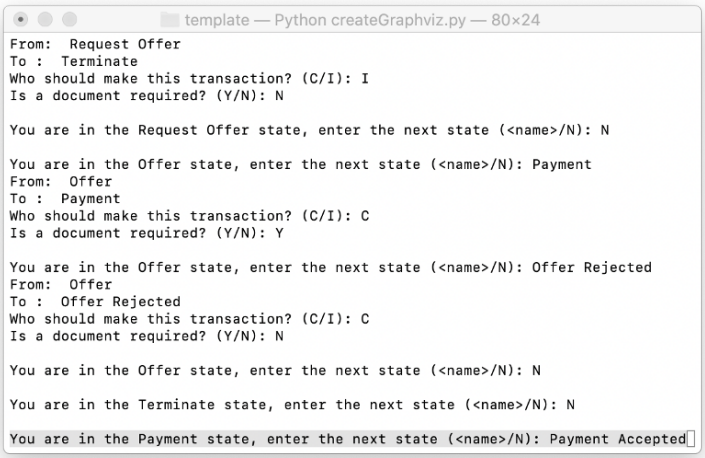

Figura 4.28. Siamo nel nodo *Payment*, aggiungiamo il nodo foglia *Payment Accepted*.

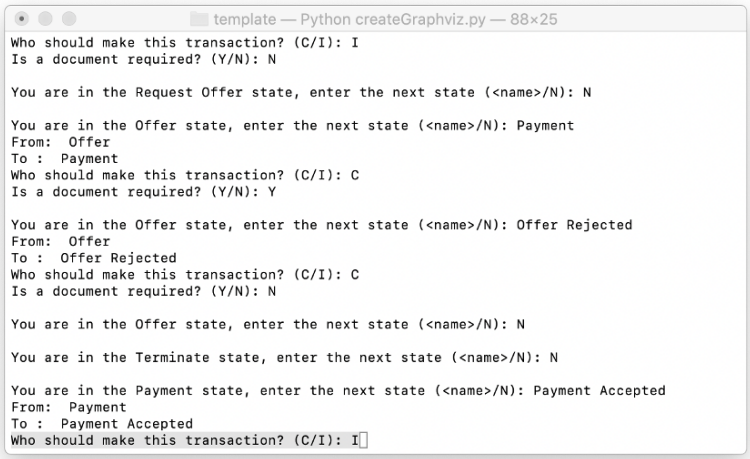

Figura 4.29. La transazione è effettuata dall'istituzione.

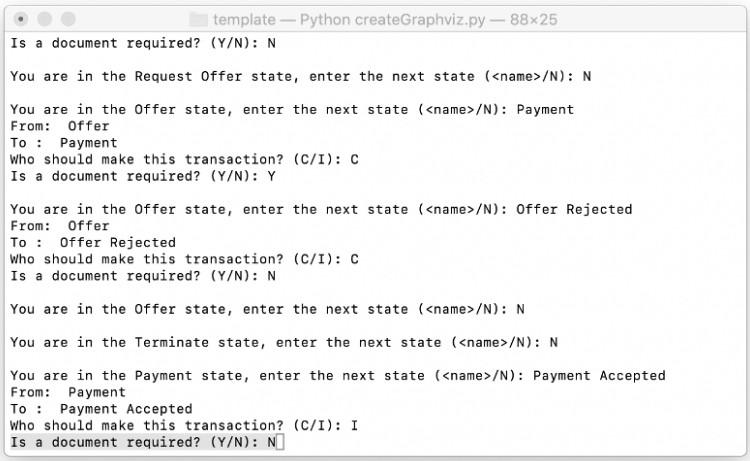

Figura 4.30. La transazione non richiede un documento allegato.

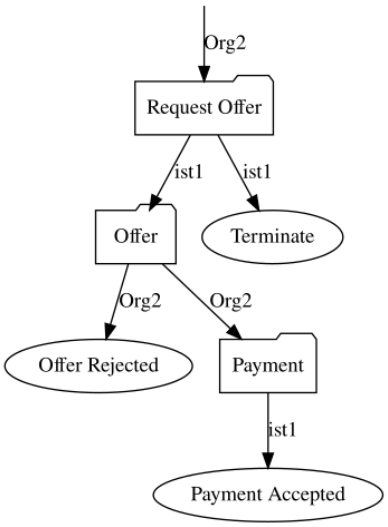

Figura 4.31. Rappresentazione grafica delle scelte precedenti.

```
template — Python createGraphviz.py — 88×25
You are in the Request Offer state, enter the next state (<name>/N): N

You are in the Offer state, enter the next state (<name>/N): Payment
From:  Offer
To :  Payment
Who should make this transaction? (C/I): C
Is a document required? (Y/N): Y

You are in the Offer state, enter the next state (<name>/N): Offer Rejected
From:  Offer
To :  Offer Rejected
Who should make this transaction? (C/I): C
Is a document required? (Y/N): N

You are in the Offer state, enter the next state (<name>/N): N

You are in the Terminate state, enter the next state (<name>/N): N

You are in the Payment state, enter the next state (<name>/N): Payment Accepted
From:  Payment
To :  Payment Accepted
Who should make this transaction? (C/I): I
Is a document required? (Y/N): N

You are in the Payment state, enter the next state (<name>/N): Payment Rejected
```

Figura 4.32. Siamo nel nodo *Payment*, aggiungiamo il nodo foglia *Payment Accepted*

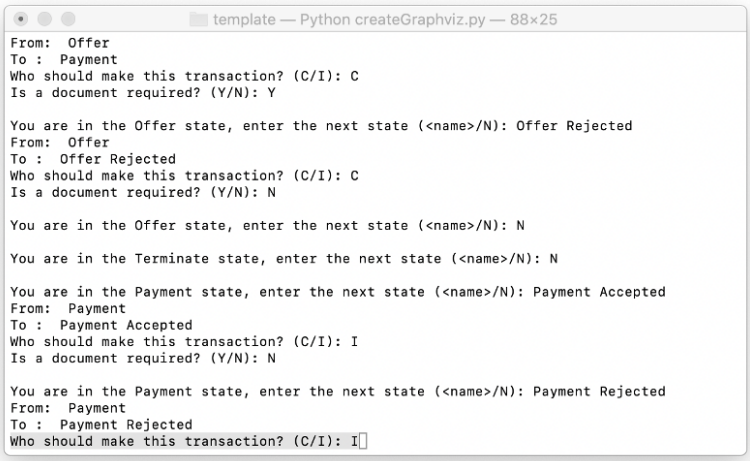

Figura 4.33. La transazione è effettuata dall'istituzione.

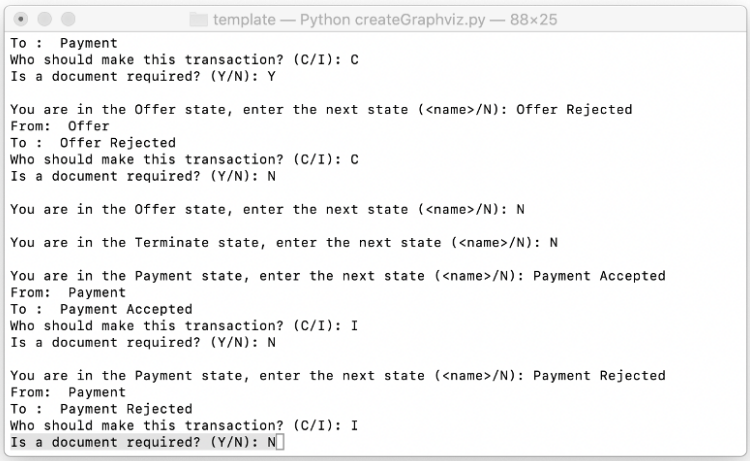

Figura 4.34. La transazione non richiede un documento allegato.

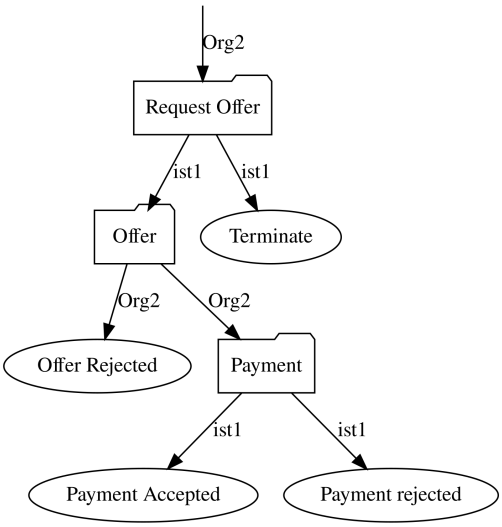

Figura 4.35. Rappresentazione grafica delle scelte precedenti.

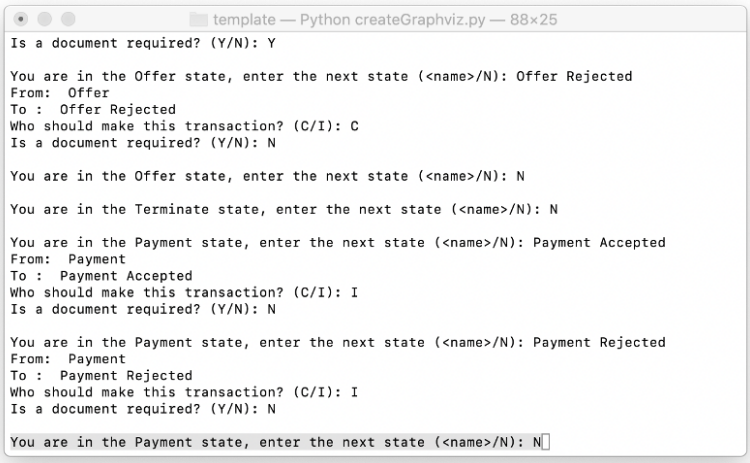

Figura 4.36. Siamo nel nodo *Payment*, non sono presenti ulteriori nodi foglia.

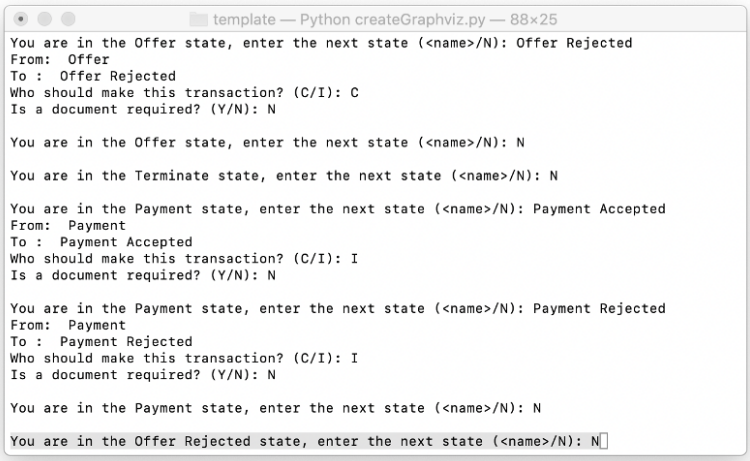

Figura 4.37. Siamo nel nodo *Offer Rejected*, non sono presenti ulteriori nodi foglia.

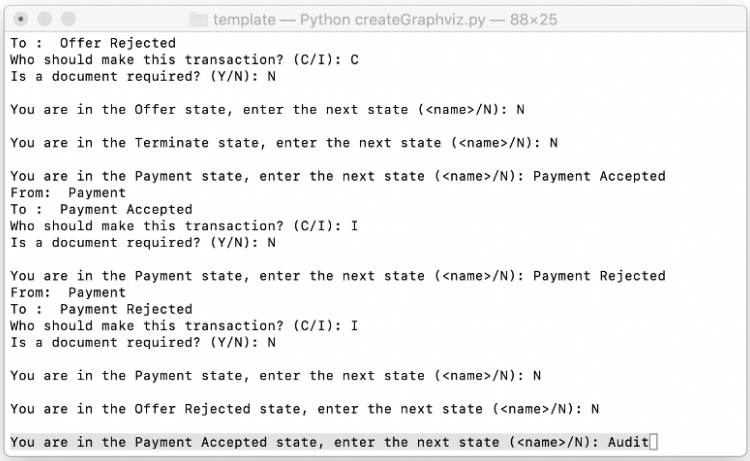

Figura 4.38. Siamo nel nodo *Paymenet Accepted*, aggiungiamo il nodo foglia *Audit*.

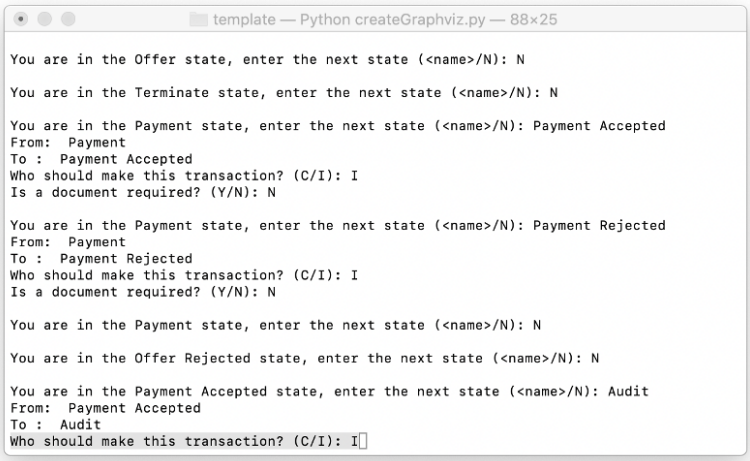

Figura 4.39. La transazione è effettuata dall'istituzione.

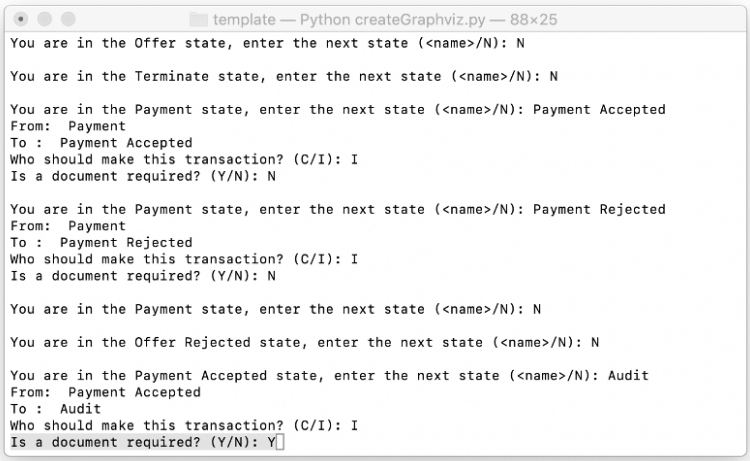

Figura 4.40. La transazione richiede un documento allegato.

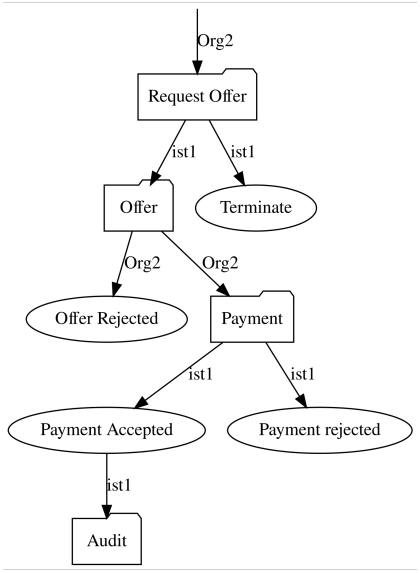

Figura 4.41. Rappresentazione grafica delle scelte precedenti.

```
template — Python createGraphviz.py — 88×25
You are in the Terminate state, enter the next state (<name>/N): N

You are in the Payment state, enter the next state (<name>/N): Payment Accepted
From:  Payment
To :  Payment Accepted
Who should make this transaction? (C/I): I
Is a document required? (Y/N): N

You are in the Payment state, enter the next state (<name>/N): Payment Rejected
From:  Payment
To :  Payment Rejected
Who should make this transaction? (C/I): I
Is a document required? (Y/N): N

You are in the Payment state, enter the next state (<name>/N): N

You are in the Offer Rejected state, enter the next state (<name>/N): N

You are in the Payment Accepted state, enter the next state (<name>/N): Audit
From:  Payment Accepted
To :  Audit
Who should make this transaction? (C/I): I
Is a document required? (Y/N): Y

You are in the Payment Accepted state, enter the next state (<name>/N): N
```

Figura 4.42. Siamo nel nodo *Payment Accepted*, non sono presenti ulteriori nodi foglia.

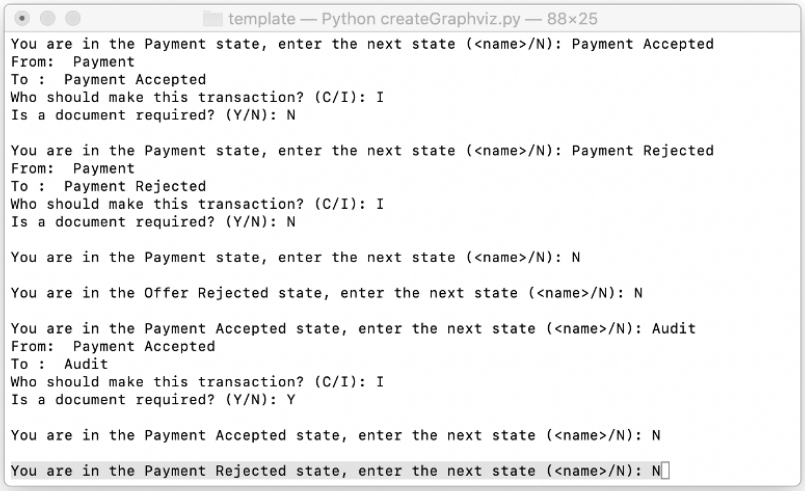

Figura 4.43. Siamo nel nodo *Payment Rejected*, non sono presenti ulteriori nodi foglia.

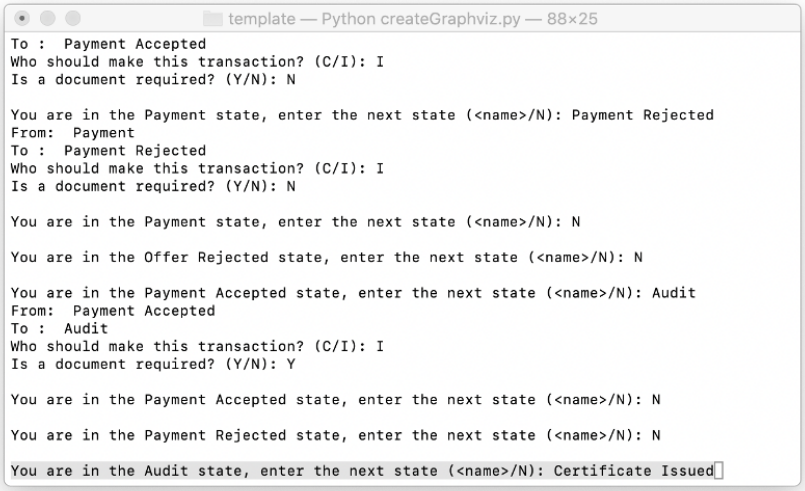

Figura 4.44. Siamo nel nodo *Audit*, aggiungiamo il nodo foglia *Certificate Issued*.

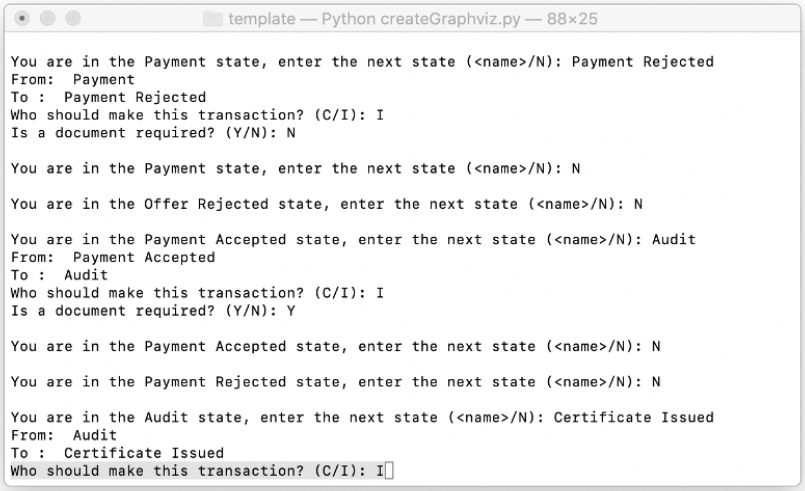

Figura 4.45. La transazione è effettuata dall'istituzione.

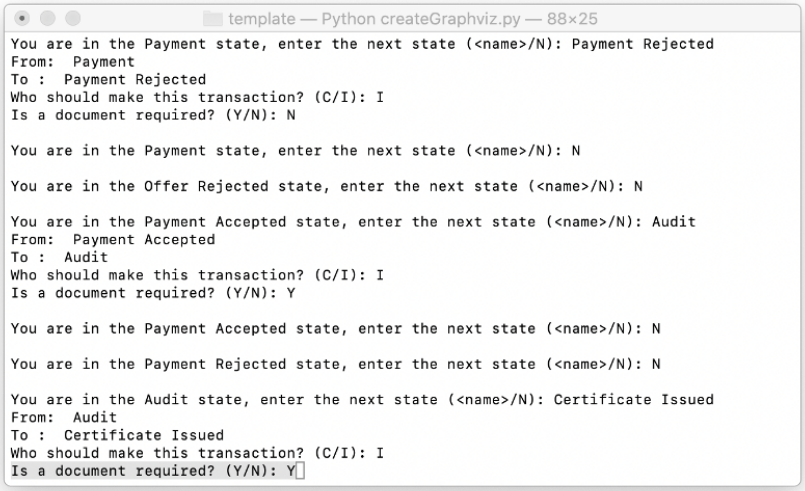

Figura 4.46. La transazione richiede un documento allegato.

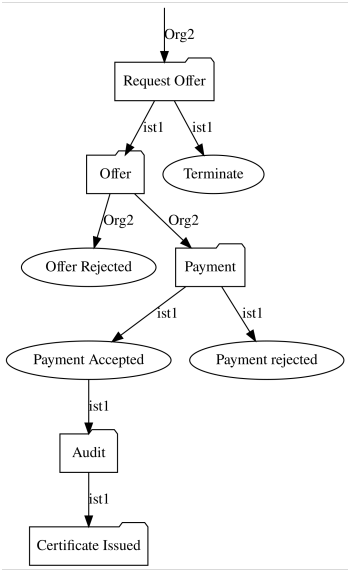

Figura 4.47. Rappresentazione grafica delle scelte precedenti.

```
template — Python createGraphviz.py — 88×25
To :  Payment Rejected
Who should make this transaction? (C/I): I
Is a document required? (Y/N): N

You are in the Payment state, enter the next state (<name>/N): N

You are in the Offer Rejected state, enter the next state (<name>/N): N

You are in the Payment Accepted state, enter the next state (<name>/N): Audit
From:  Payment Accepted
To :  Audit
Who should make this transaction? (C/I): I
Is a document required? (Y/N): Y

You are in the Payment Accepted state, enter the next state (<name>/N): N

You are in the Payment Rejected state, enter the next state (<name>/N): N

You are in the Audit state, enter the next state (<name>/N): Certificate Issued
From:  Audit
To :  Certificate Issued
Who should make this transaction? (C/I): I
Is a document required? (Y/N): Y

You are in the Audit state, enter the next state (<name>/N): Certificate Rejected
```

Figura 4.48. Siamo nel nodo *Audit*, aggiungiamo il nodo foglia *Certificate Rejected*.

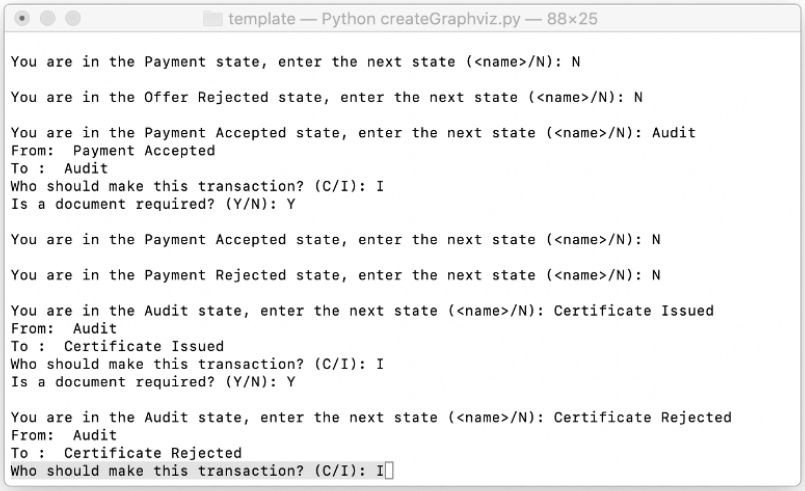

Figura 4.49. La transazione è effettuata dall'istituzione.

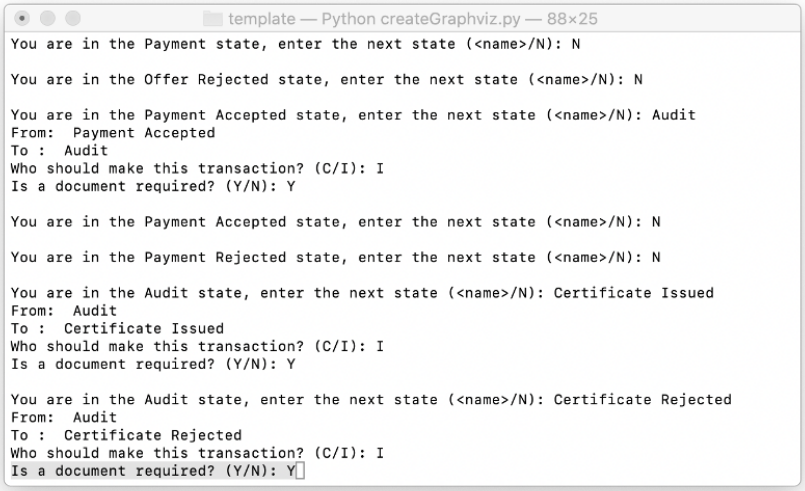

Figura 4.50. La transazione richiede un documento allegato.

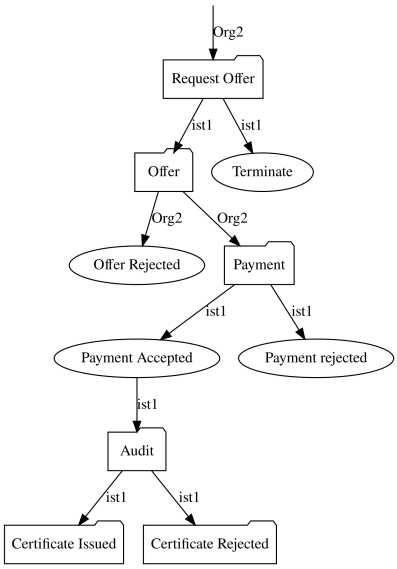

Figura 4.51. Rappresentazione finale della macchina a stati.

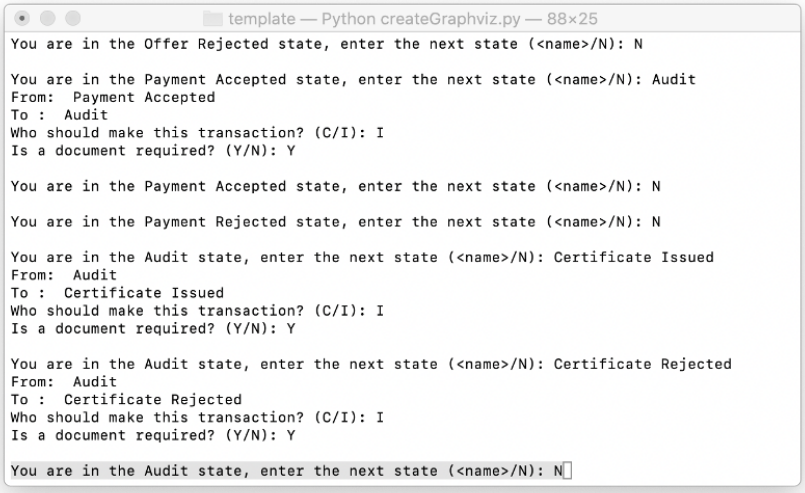

Figura 4.52. Siamo nel nodo *Audit*, non sono presenti ulteriori nodi foglia.

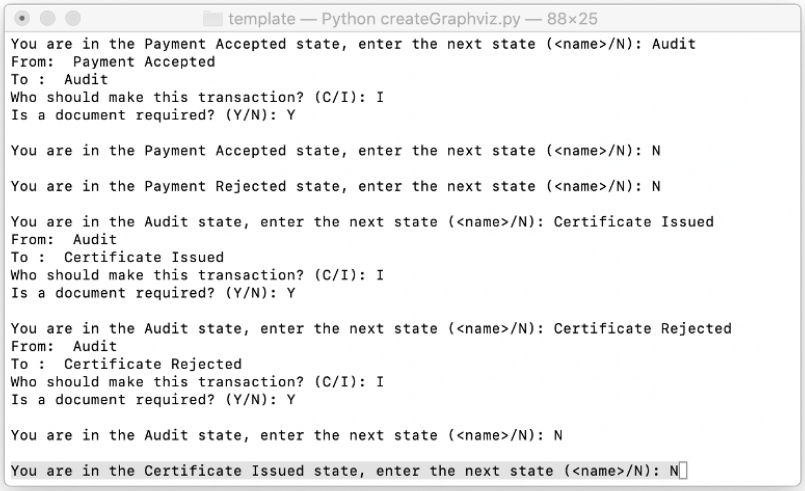

Figura 4.53. Siamo nel nodo *Certificate Issued*, non sono presenti ulteriori nodi foglia.

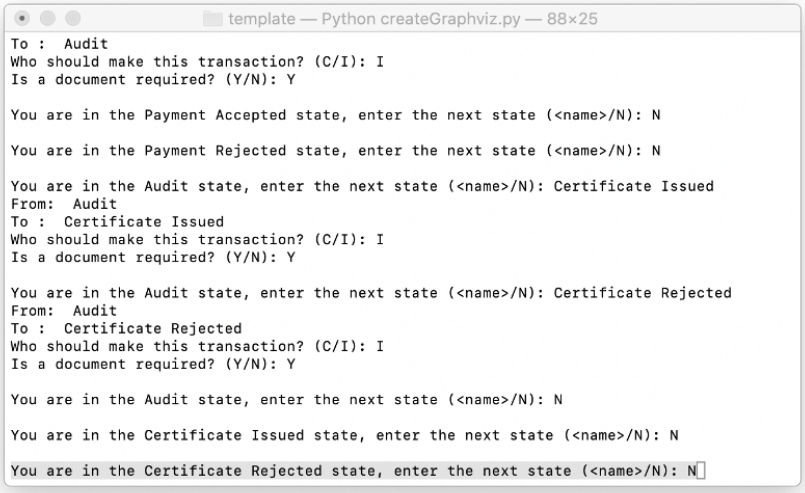

Figura 4.54. Siamo nel nodo *Certificate Rejected*, non sono presenti ulteriori nodi foglia.

Figura 4.55. Terminata la visita in profondità il file viene creato automaticamente.

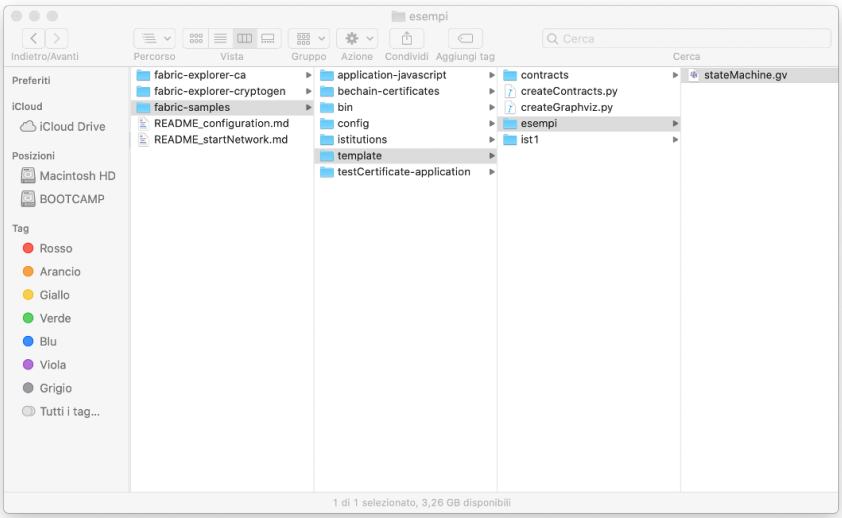

Figura 4.56. Creazione del file avvenuta con successo.

stateMachine.gv

Free Mode

/Applications/XAMPP/xamppfiles/htdocs/Hyperledger/fabric-samples/template/esempi/stateMachine.gv

```
digraph G {

      Start [label="" shape=none,height=.0,width=.0]
      "Request Offer" [id="Request Offer" shape=folder]
      "Offer" [id="Offer" shape=folder]
      "Terminate" [id="Terminate" shape=ellipse]
      "Payment" [id="Payment" shape=folder]
      "Offer Rejected" [id="Offer Rejected" shape=ellipse]
      "Payment Accepted" [id="Payment Accepted" shape=ellipse]
      "Payment Rejected" [id="Payment Rejected" shape=ellipse]
      "Audit" [id="Audit" shape=folder]
      "Certificate Issued" [id="Certificate Issued" shape=folder]
      "Certificate Rejected" [id="Certificate Rejected" shape=folder]

      Start -> "Request Offer" [label = "Org2"]
      "Request Offer" -> "Offer" [label = "ist1"]
      "Request Offer" -> "Terminate" [label = "ist1"]
      "Offer" -> "Payment" [label = "Org2"]
      "Offer" -> "Offer Rejected" [label = "Org2"]
      "Payment" -> "Payment Accepted" [label = "ist1"]
      "Payment" -> "Payment Rejected" [label = "ist1"]
      "Payment Accepted" -> "Audit" [label = "ist1"]
      "Audit" -> "Certificate Issued" [label = "ist1"]
      "Audit" -> "Certificate Rejected" [label = "ist1"]

 }
```

L: 30 C: 2 (none) Saved: 16/05/2022, 15:50:32 1.161 / 119 / 30 - 100%

Figura 4.57. Contenuto del file *stateMachine.gv*.

4.3.2 Creazione di un chaincode

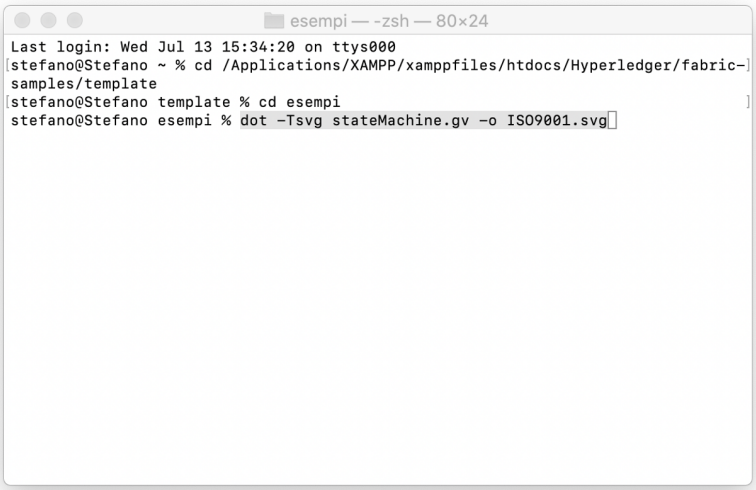

Figura 4.58. Generiamo il file *ISO9001.svg*.

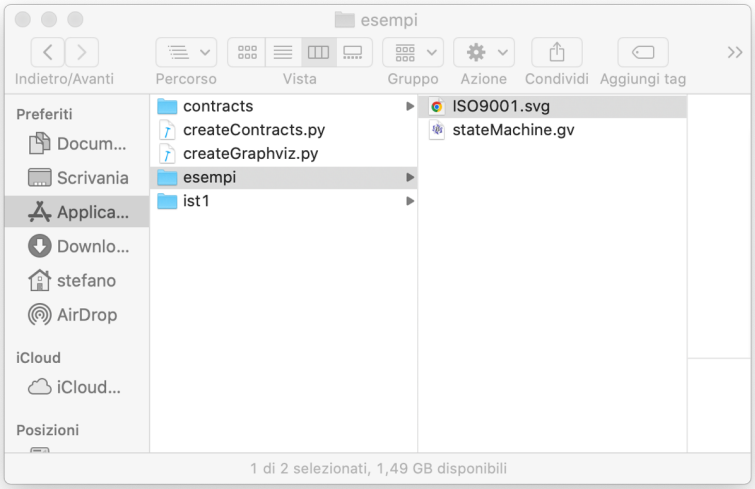

Figura 4.59. File generato correttamente.

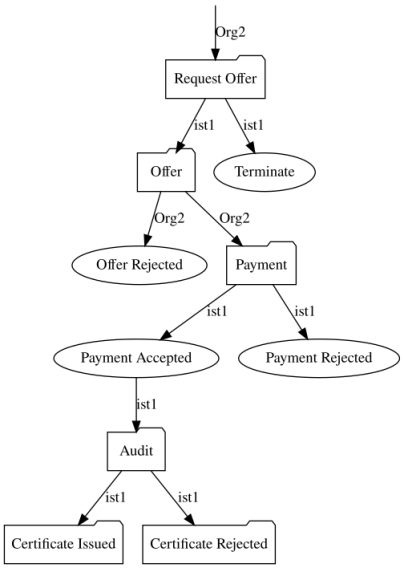

Figura 4.60. Contenuto del file *ISO9001.svg*.

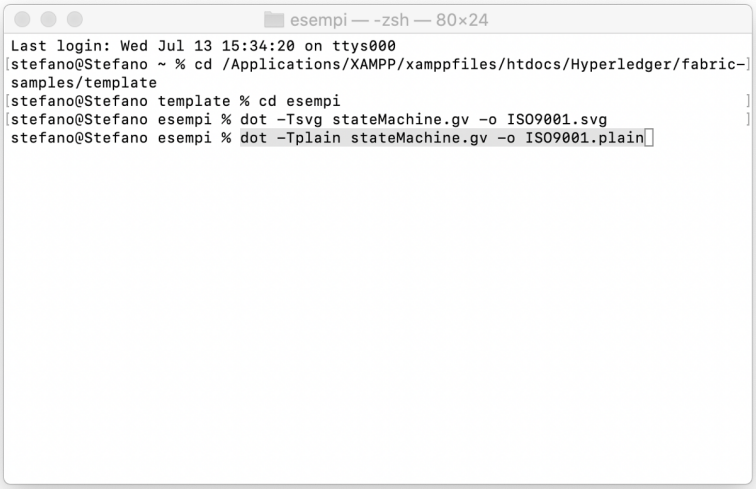

Figura 4.61. Generiamo il file *ISO9001.plain*.

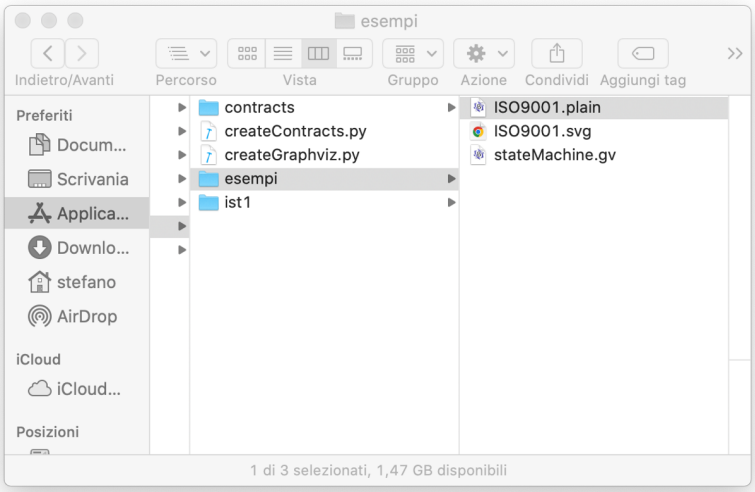

Figura 4.62. File generato correttamente.

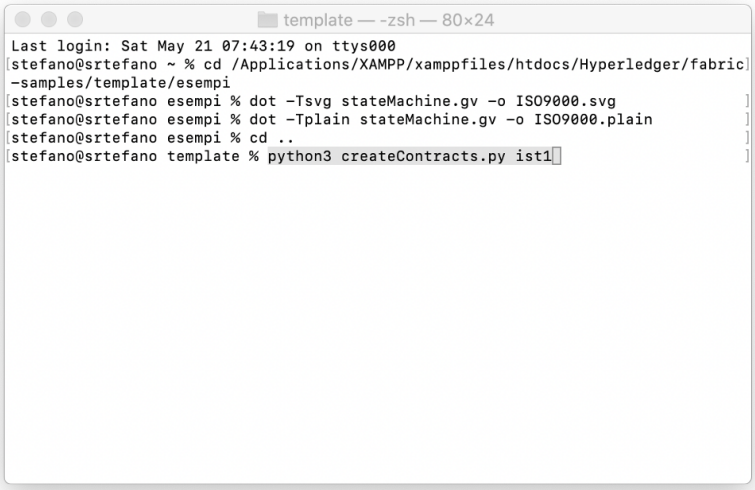

Figura 4.63. Eseguiamo lo script python *createContracts.py* passando come parametro il nome dell'istituzione a cui si riferisce. In questo caso il nome è *ist1*.

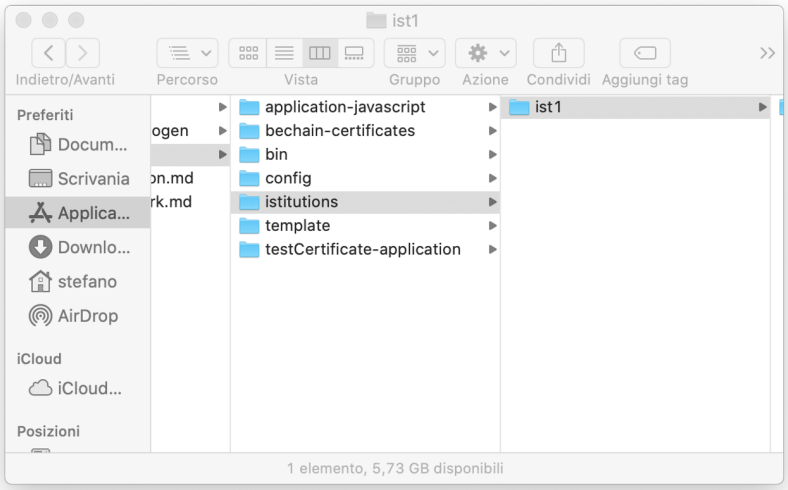

Figura 4.64. Chaincode generato correttamente.

Capitolo 5

Risultati sperimentali

5.1 Applicazione

La comunicazione tra gli enti di certificazioni e i rispettivi clienti è resa possibile grazie a un'interfaccia web. In questo capitolo dedicato all'applicazione web, vengono mostrate le diverse funzionalità del programma a fronte delle scelte progettuali esposte nel terzo capitolo.

Le entità coinvolte in questo esempio sono quattro: da un lato gli enti di certificazione *ist1* e *ist2*, dall'altro i clienti *polito* e *unito* che intendono ottenere un certificato. In particolare verranno mostrate, passo dopo passo, le seguenti funzionalità: registrazione alla rete blockchain, invito di un cliente, inizializzazione di un processo di certificazione, esecuzione di una transazione, annullamento di una transazione, visualizzazione delle transazioni passate relative a un determinato oggetto aziendale, ricerca sulla base del cliente, ricerca sulla base del certificato e ricerca sulla base di chiavi multiple.

Per concludere, con lo scopo di dimostrare l'effettiva inviolabilità dello *smart contract*, si forzeranno una serie di azioni non valide mostrando di volta in volta i messaggi di errore risultanti.

5.1.1 Funzionalità

Registrazione di un'istituzione

Per registrarsi come istituzione è necessario cliccare *Signup*, completare i diversi campi del form e infine cliccare *Register*.

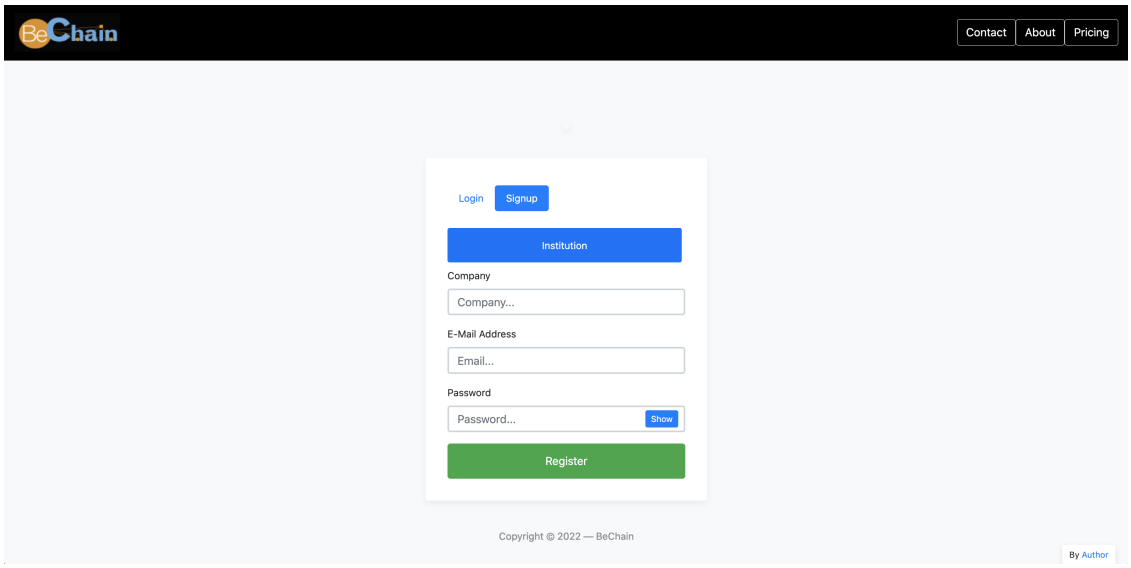

Figura 5.1. Form di registrazione.

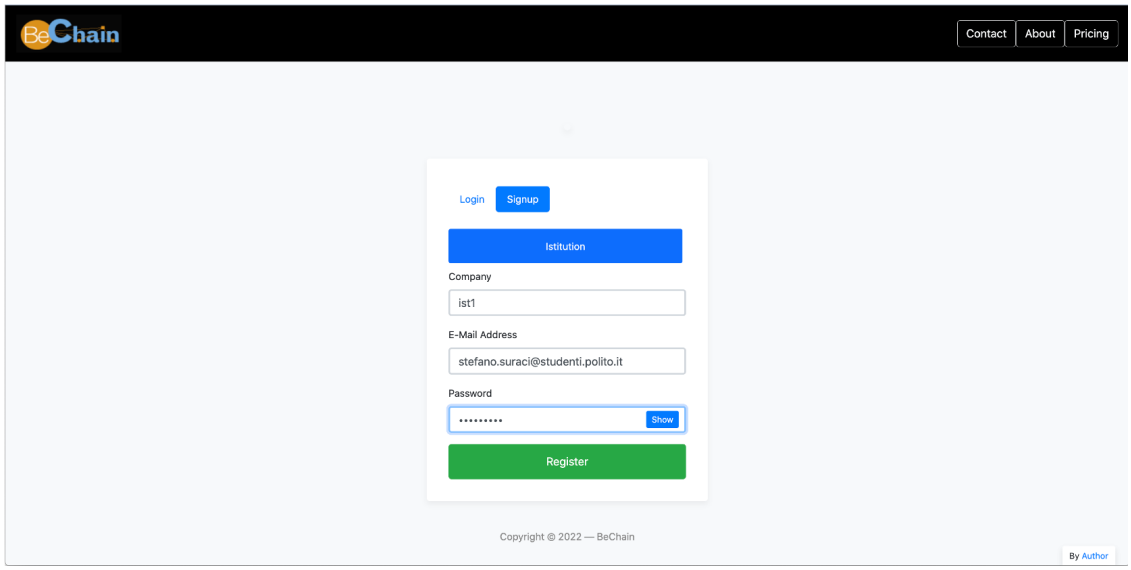

Figura 5.2. Form di registrazione completato con i dati dell'istituzione.

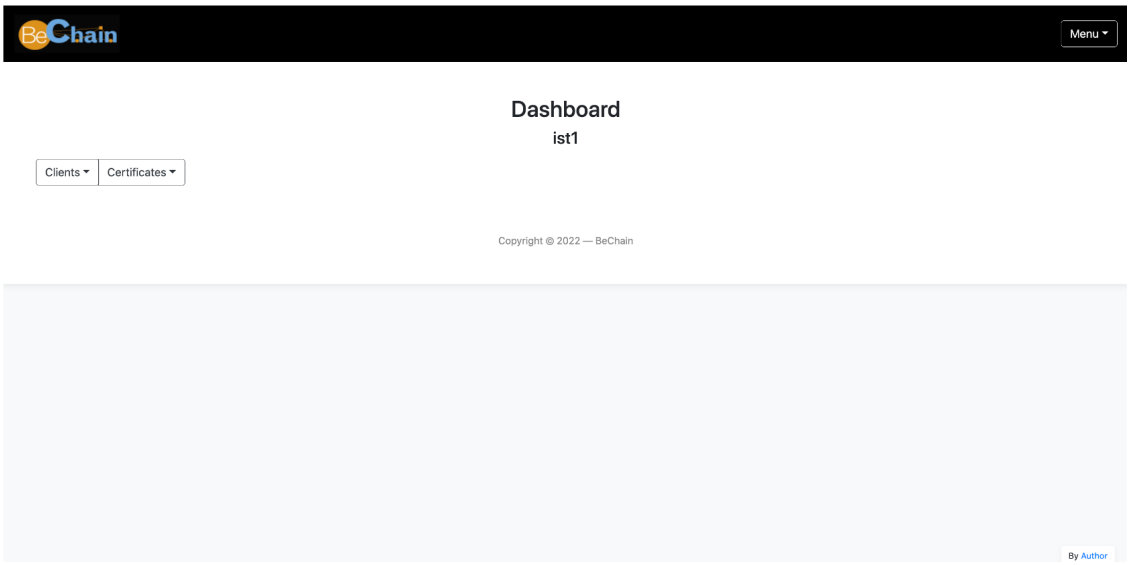

Figura 5.3. Se la registrazione è avvenuta con successo si viene indirizzati automaticamente alla propria pagina personale.

Invito di un cliente

Solo le istituzioni possono invitare nuovi clienti. Questo paragrafo mostra come l'*ist1* invia un codice di invito attraverso la sua pagina personale. Il codice di invito, ricevuto tramite email dal cliente, può essere usato una sola volta e scade automaticamente dopo ventiquattro ore.

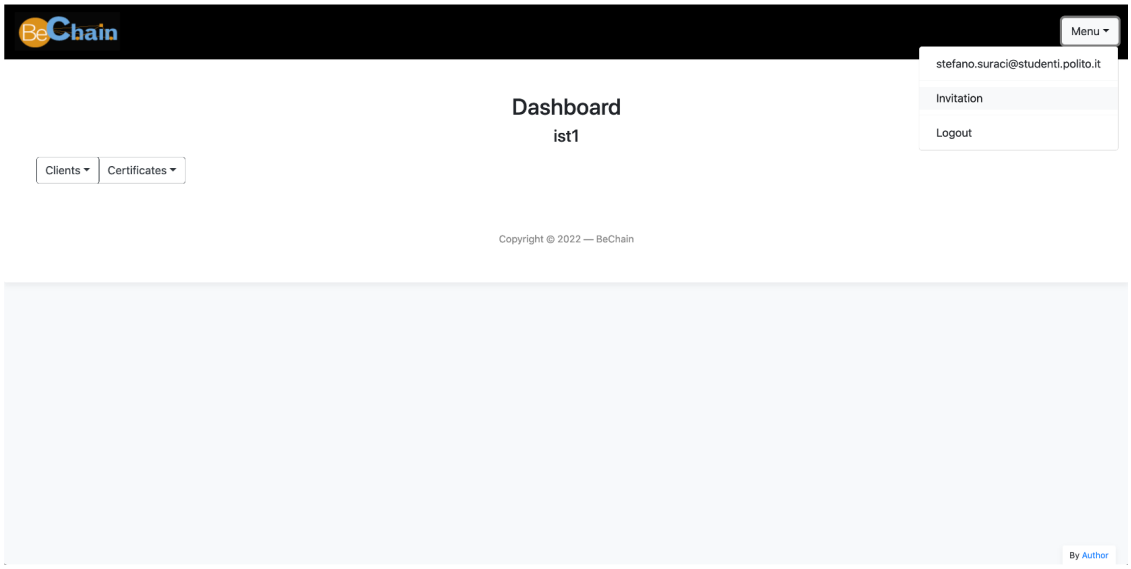

Figura 5.4. Clicchiamo su *Menù* e successivamente su *Invitation*.

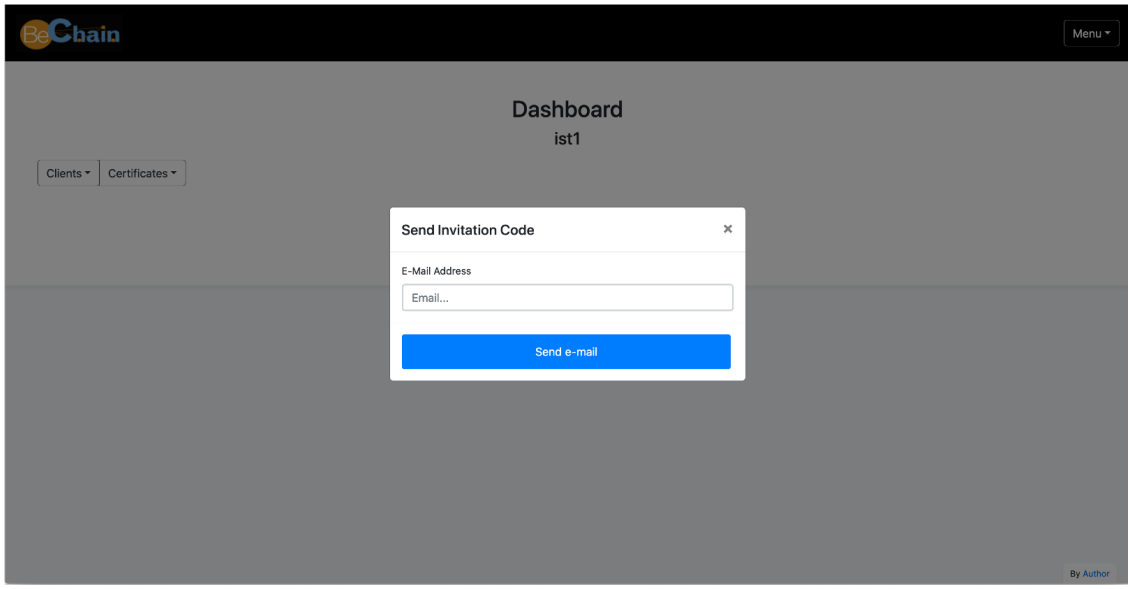

Figura 5.5. Form codice di invito.

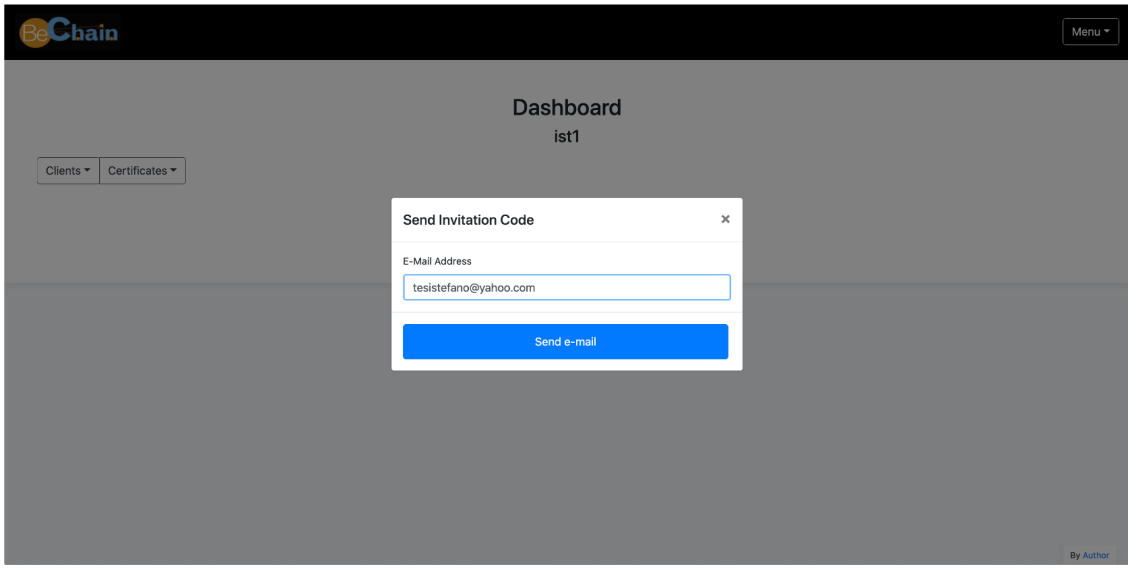

Figura 5.6. Form codice di invito completato con l'email del cliente.

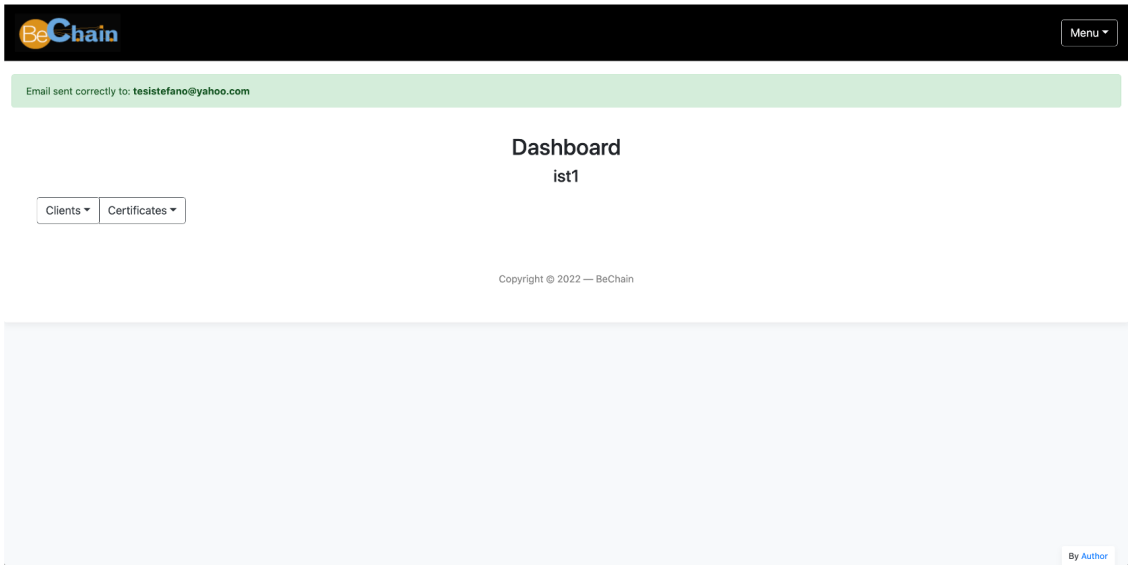

Figura 5.7. *Alert* di successo: email inviata correttamente.

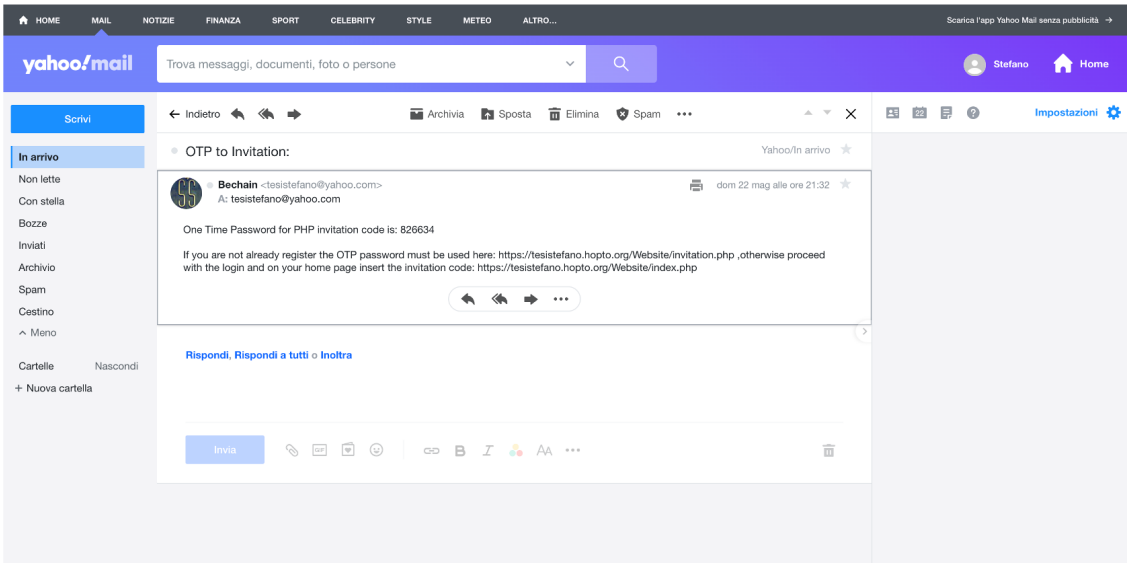

Figura 5.8. Codice di invito ricevuto tramite la casella di posta del cliente. Cliccando sul link è possibile essere indirizzati automaticamente alla pagina di invito del cliente.

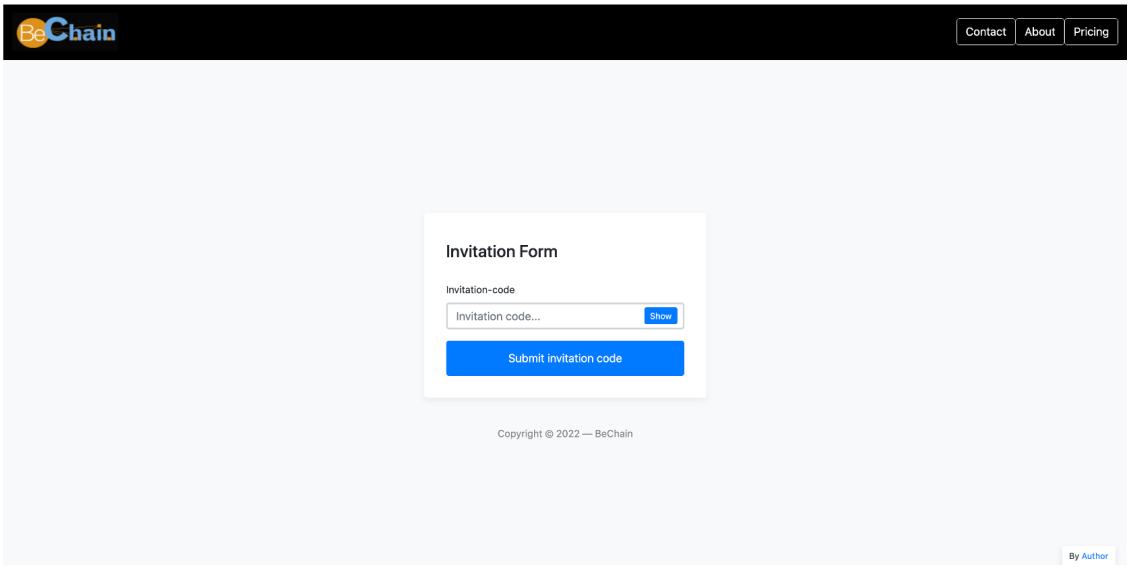

Figura 5.9. Form di invito.

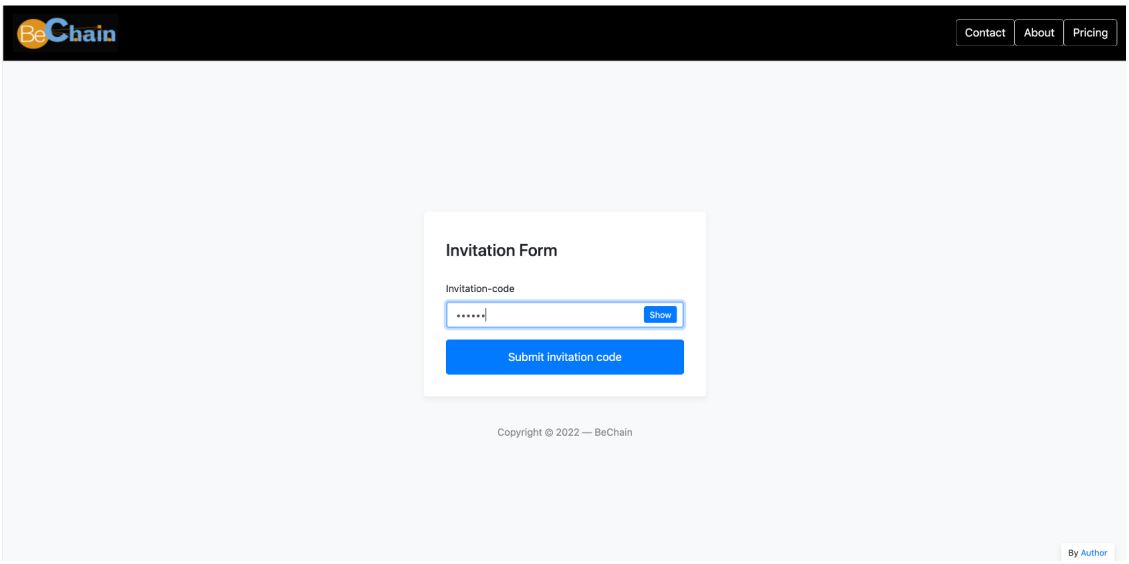

Figura 5.10. Form di invito completato con la password monouso ricevuta per email.

Registrazione di un cliente

Se il codice di invito inserito è corretto, si viene indirizzati automaticamente alla pagina di registrazione del cliente. È bene sottolineare che questa è l'unica possibile procedura per poter completare la registrazione come clienti, cioè solo se si è stati direttamente invitati da un'istituzione.

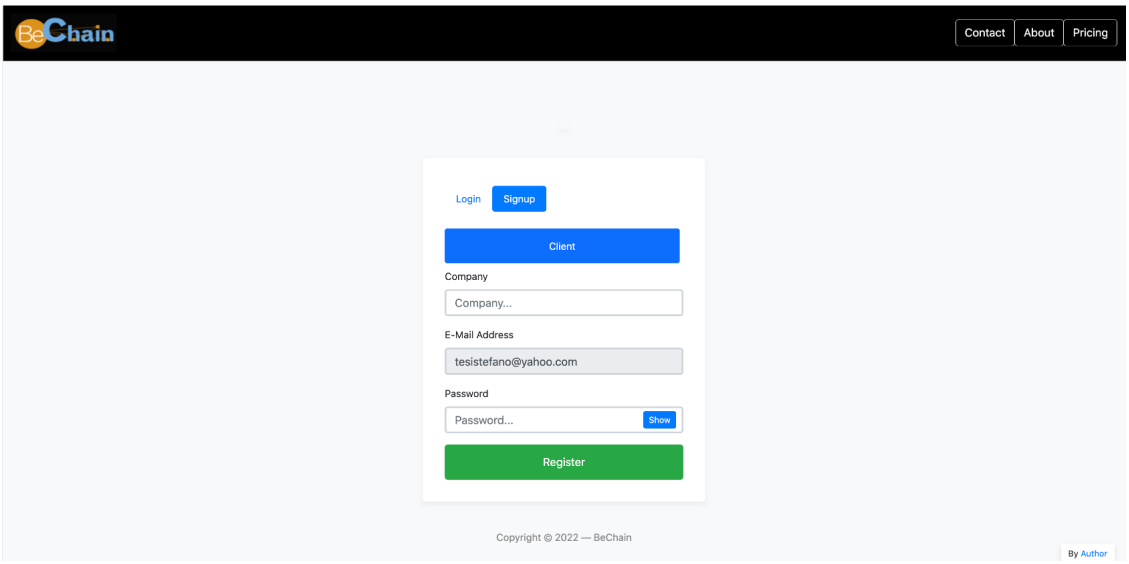

Figura 5.11. Form di registrazione del cliente. L'email è assegnata automaticamente e corrisponde a quella usata durante la procedura di invito.

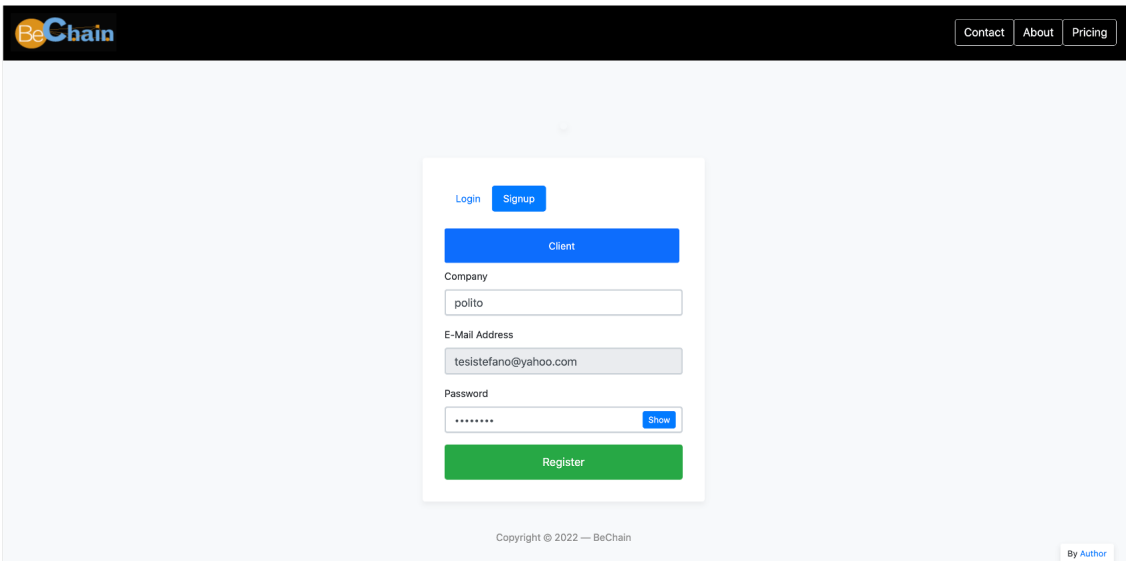

Figura 5.12. Form di registrazione del cliente completato.

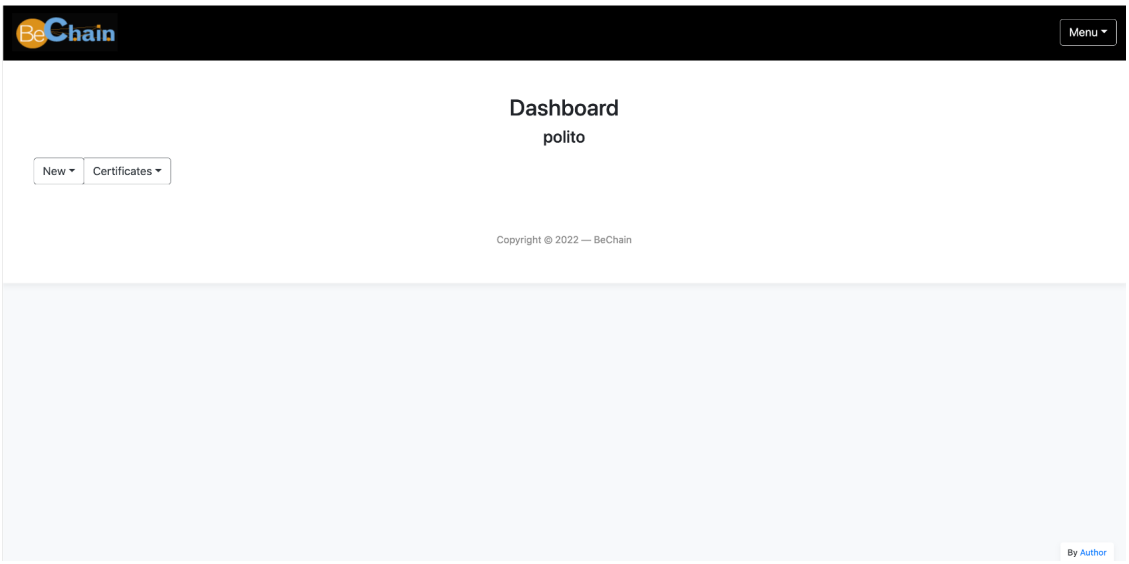

Figura 5.13. Pagina personale del cliente *polito*.

Avvio di un processo di certificazione

Completata la registrazione, *polito* è a tutti gli effetti cliente dell'*ist1*. A questo punto può avviare un processo di certificazione per uno qualsiasi dei certificati offerti dall'istituzione.

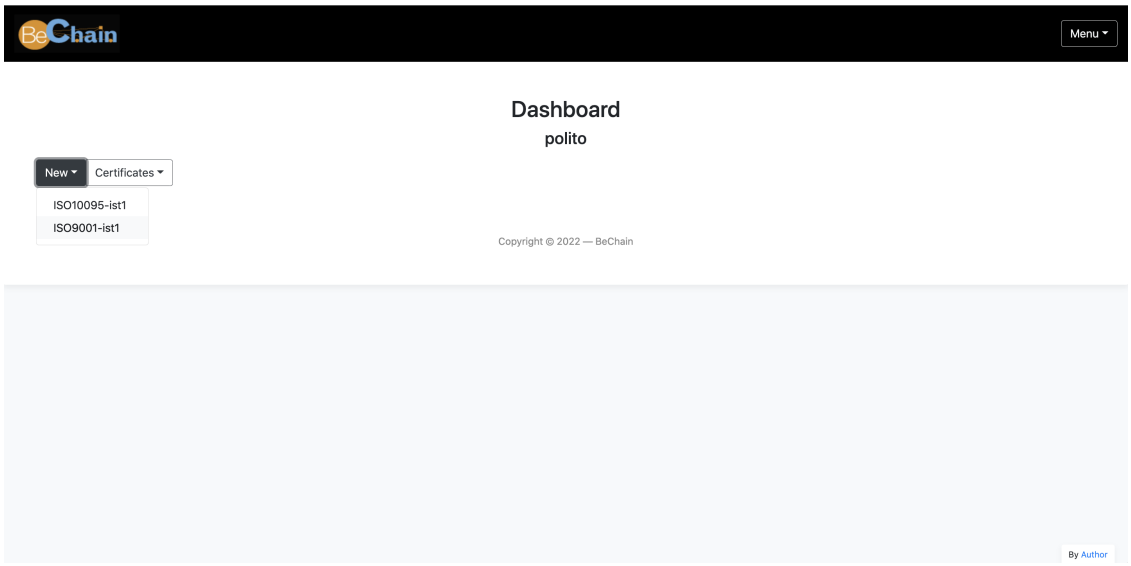

Figura 5.14. Cliccando su *Menù* viene mostrato l'elenco dei certificati offerti dall'*ist1*. Successivamente clicchiamo su *ISO9001-ist1*.

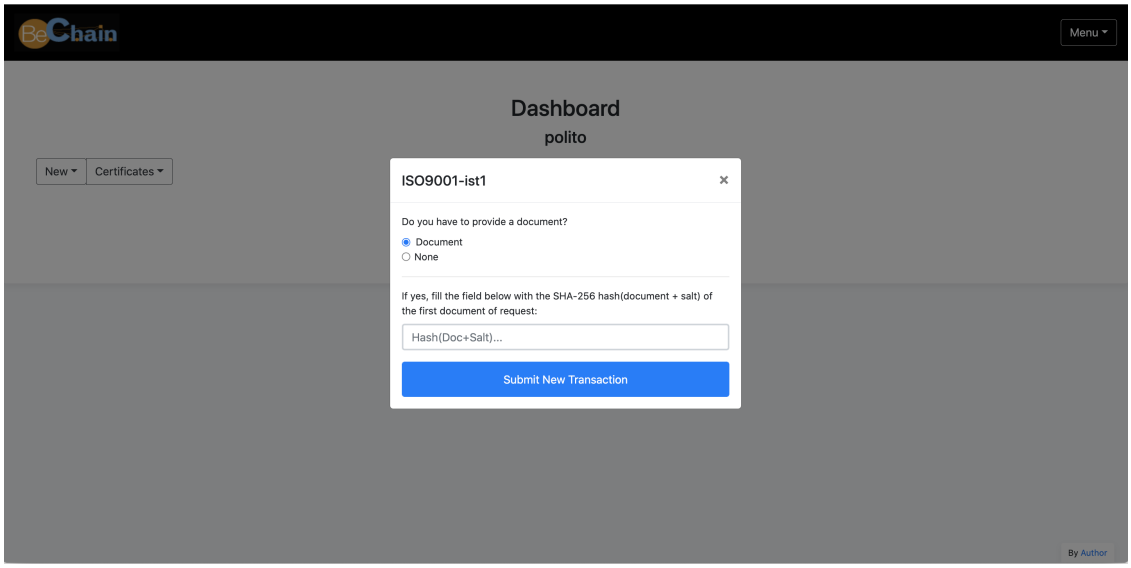

Figura 5.15. Form nuova transazione. I campi del form richiedono di selezionare la presenza o meno di un documento e di digitare nel caso il suo hash più il sale.

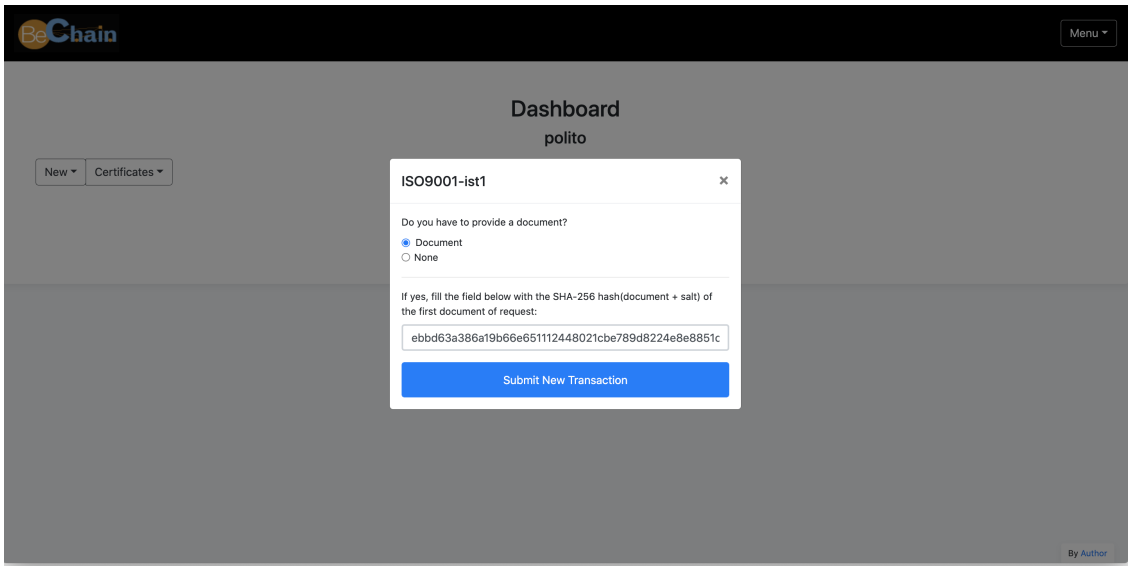

Figura 5.16. Selezioniamo la voce *Document* e digitiamo l'hash più il sale del documento.

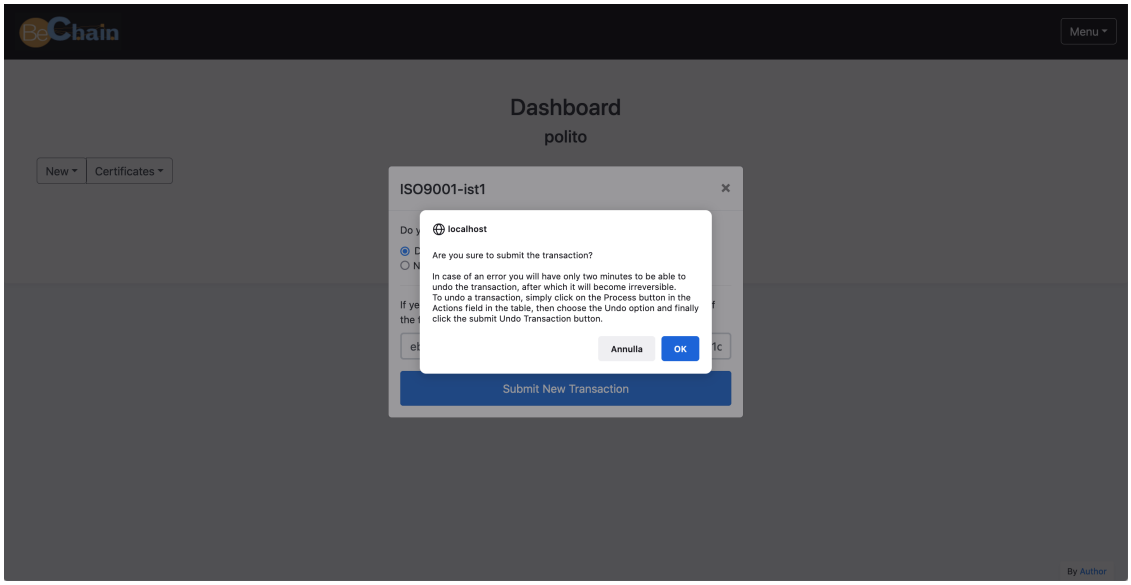

Figura 5.17. Messaggio di allerta. Questo messaggio avvisa l'utente che dopo due minuti la transazione diventa irreversibile. Si tratta quindi di un invito a controllare che i dati inseriti siano corretti.

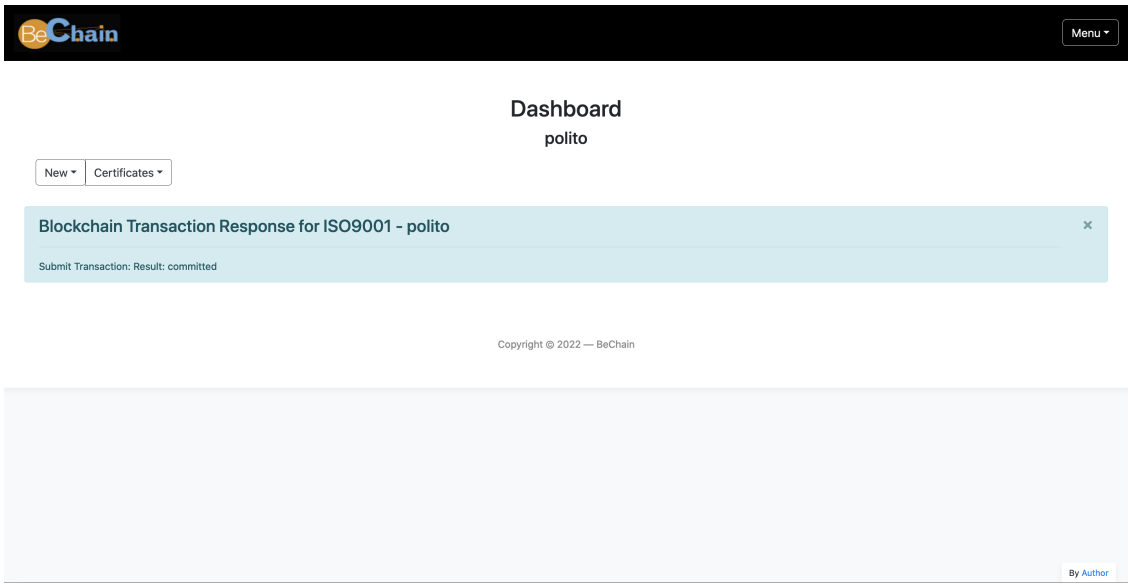

Figura 5.18. Messaggio di avviso. La transazione è stata eseguita con successo.

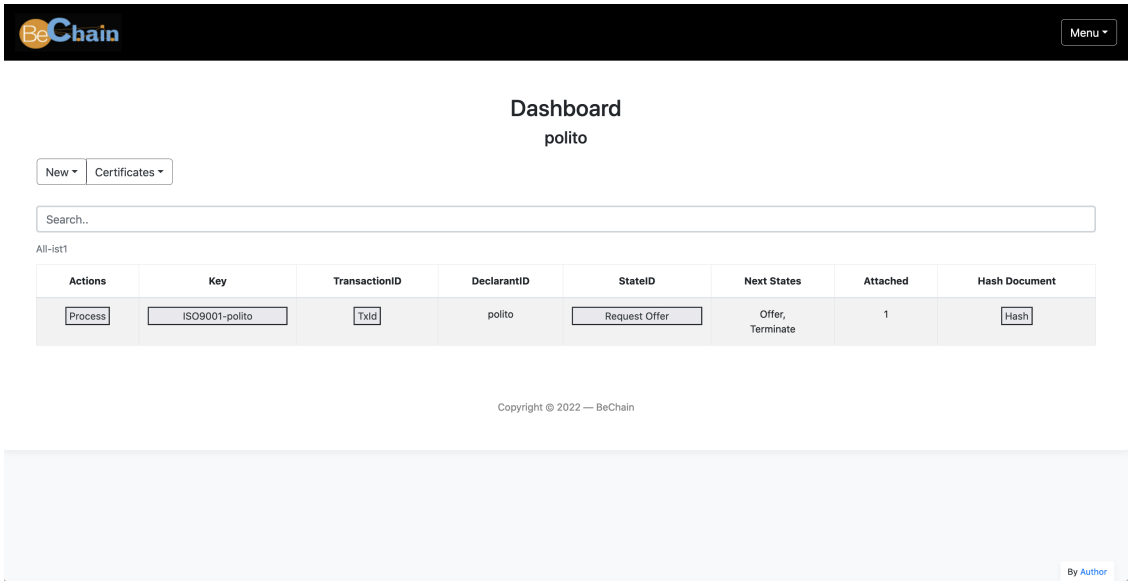

Figura 5.19. L'ultimo stato della transazione è visibile nella pagina personale, quest'ultima è l'interfaccia verso lo stato mondiale dell'istituzione.

Visualizzazione dell'hash del documento

In questo paragrafo viene mostrato come visualizzare l'hash del documento. Quest'ultimo è stato fornito al passo precedente dal cliente durante l'esecuzione della transazione.

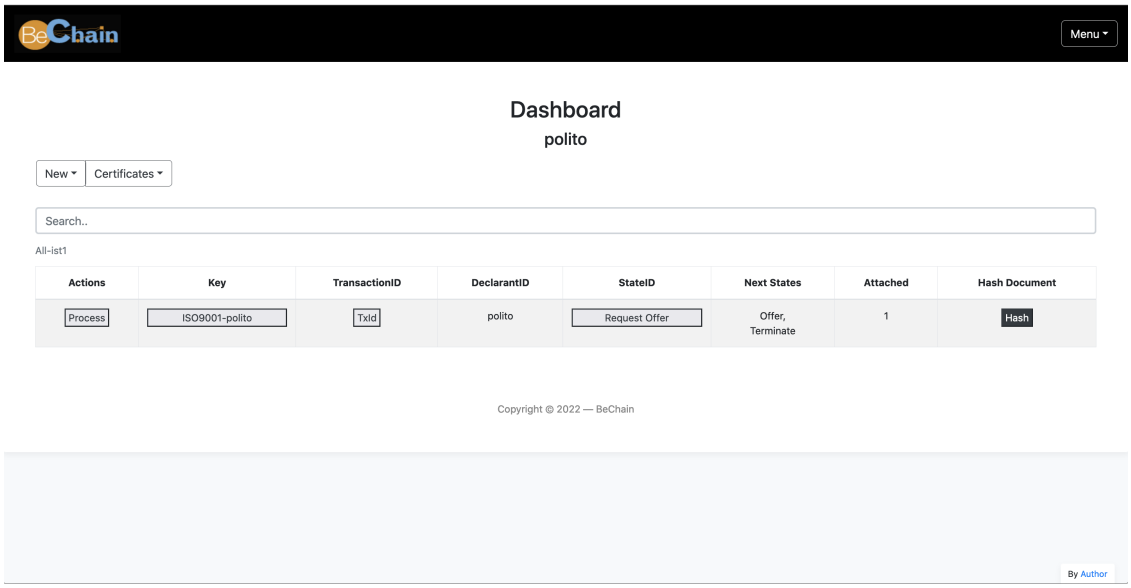

Figura 5.20. Clicchiamo sul pulsante *Hash*.

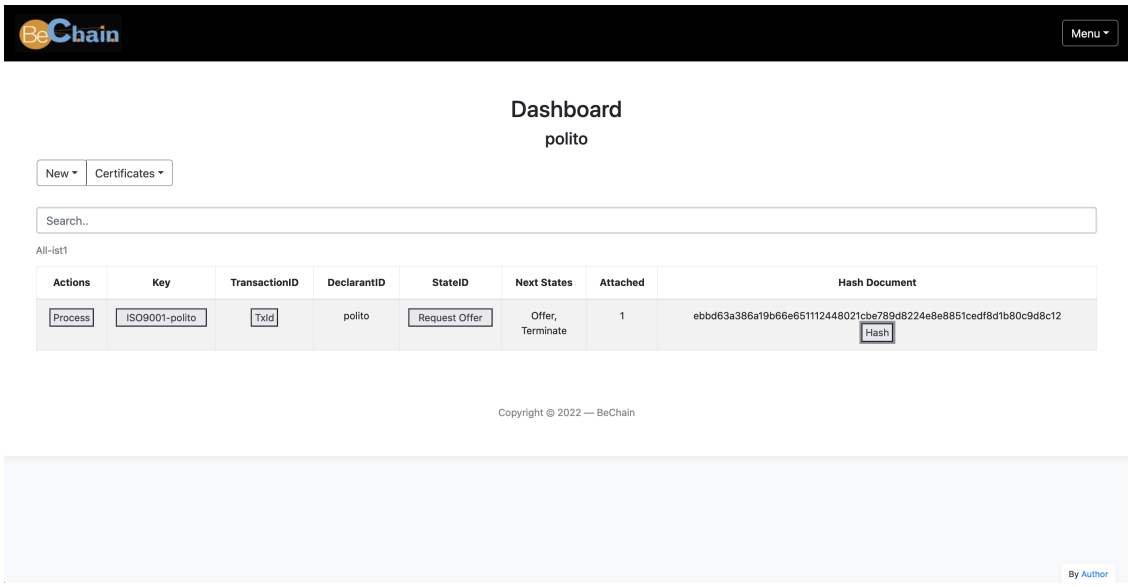

Figura 5.21. Visualizzazione dell'hash del documento.

Visualizzazione dell'id della transazione

In questo paragrafo viene mostrato come visualizzare l'id associato alla transazione nella blockchain.

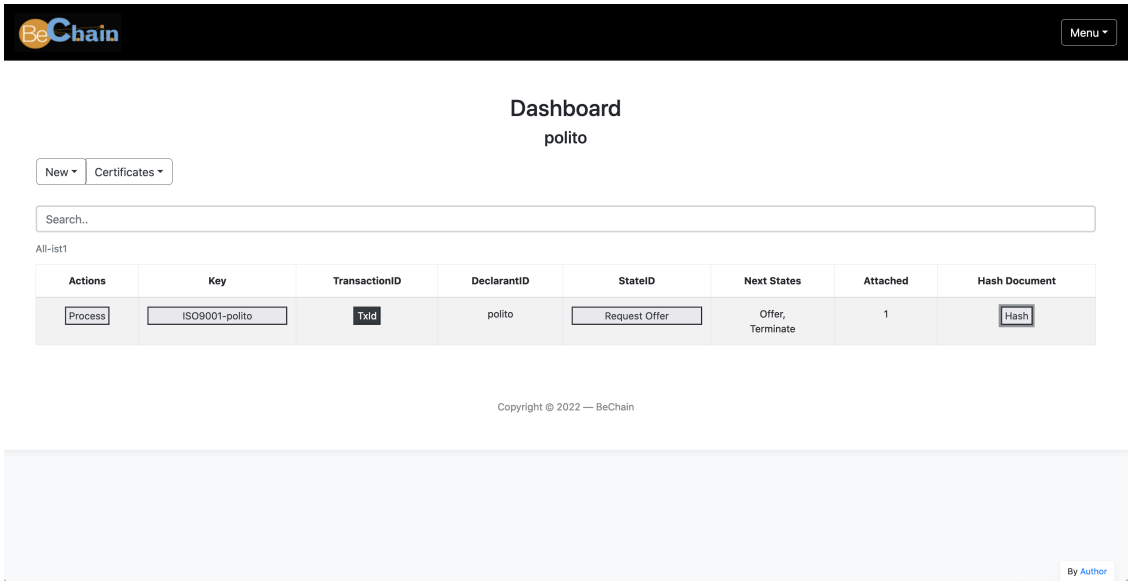

Figura 5.22. Clicchiamo su *TxId*.

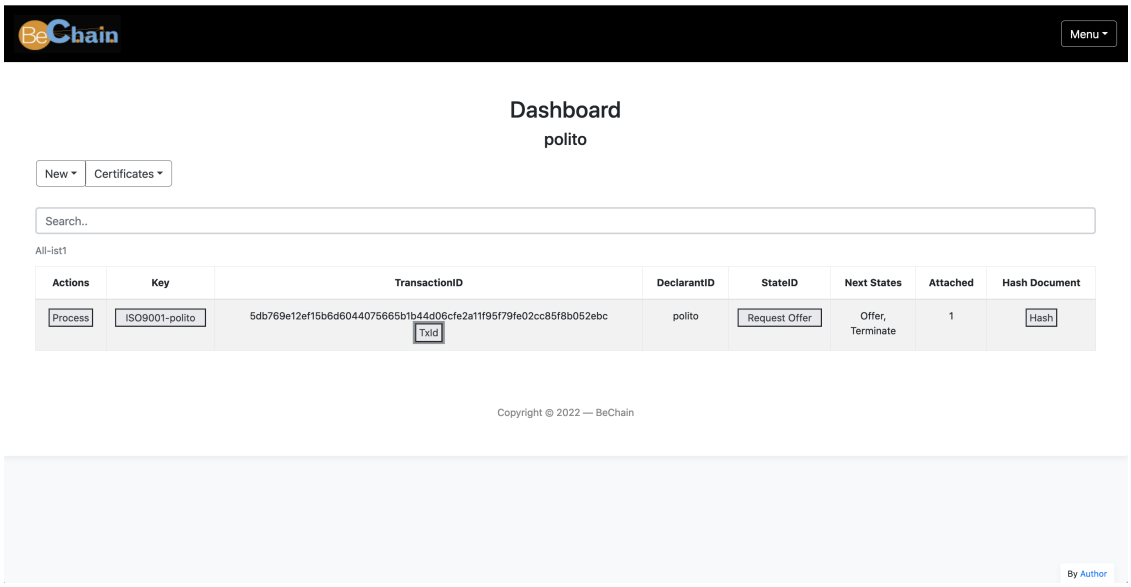

Figura 5.23. Visualizzazione dell'id della transazione.

Visualizzazione dello stato corrente

Ogni nuova transazione comporta un cambio di stato nel processo di certificazione. In questo paragrafo viene mostrato come visualizzare graficamente lo stato attuale all'interno della macchina a stati.

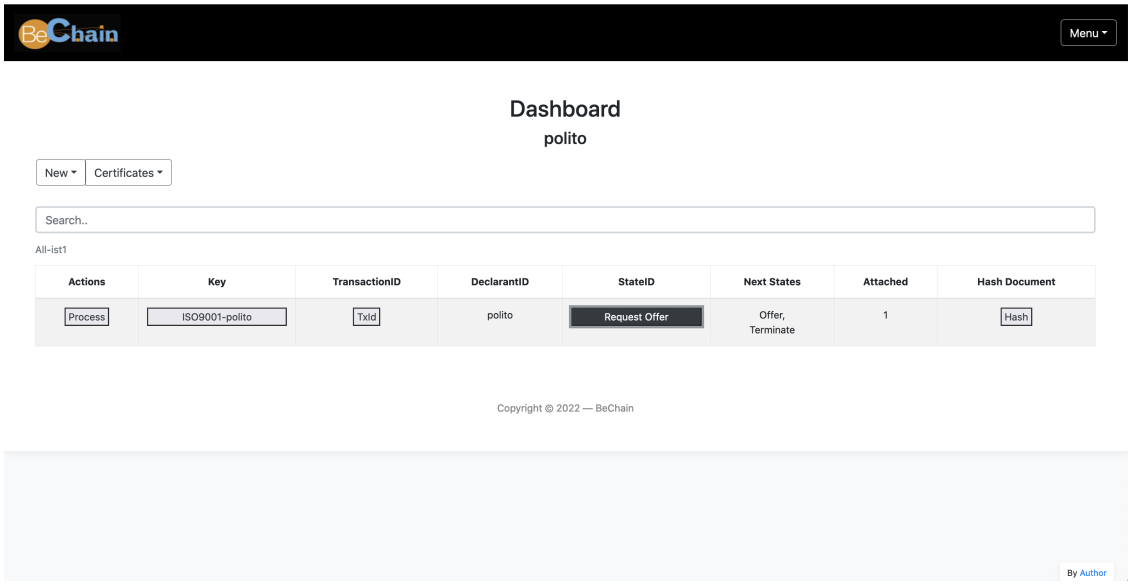

Figura 5.24. Clicchiamo su *Request Offer*.

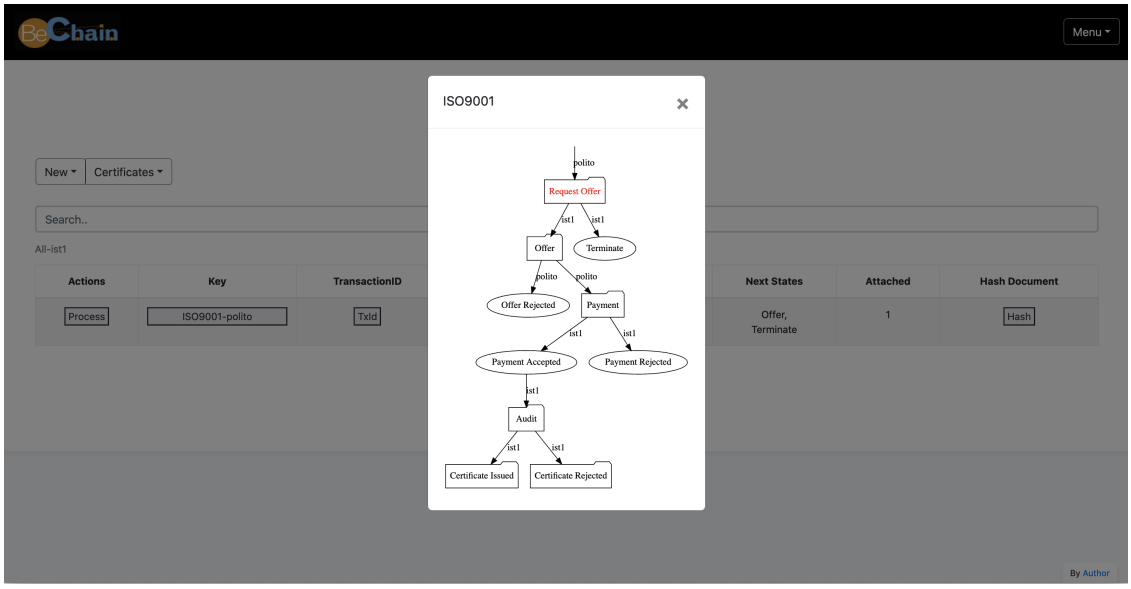

Figura 5.25. Visualizzazione della macchina a stati relativa al processo di certificazione ISO9001. Lo stato corrente è evidenziato in rosso.

Esecuzione di una nuova transazione

In questo paragrafo agendo in qualità dell'*ist1* eseguiamo una nuova transazione relativamente al processo di certificazione ISO9001 avviato da *polito*.

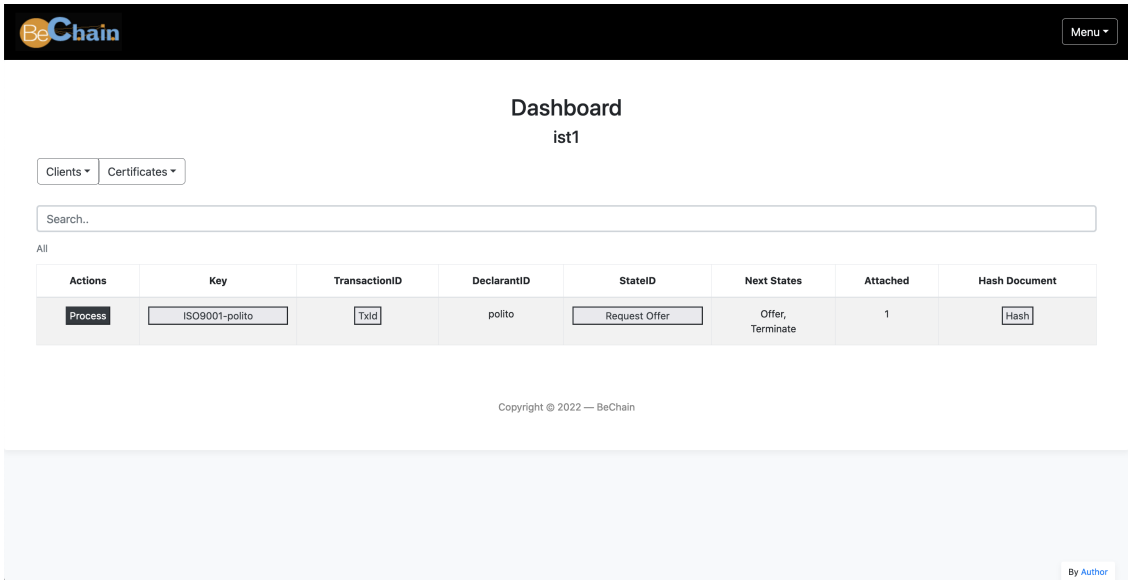

Figura 5.26. Per eseguire una nuova transazione, clicchiamo sul pulsante *Process*.

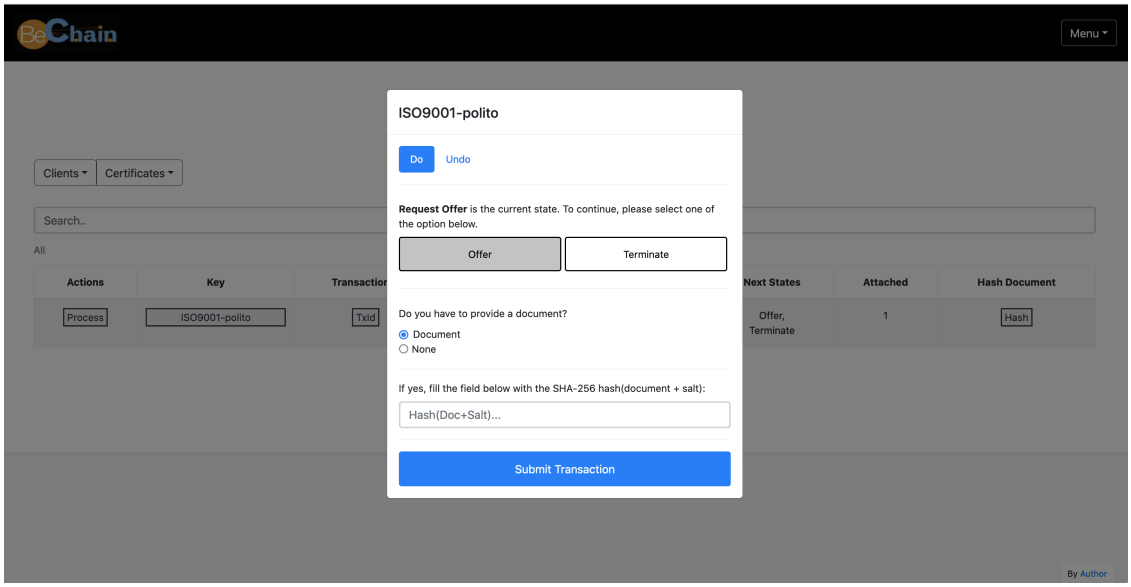

Figura 5.27. I prossimi due stati del processo di certificazione sono *Offer* e *Terminate*. Selezioniamo *Offer*.

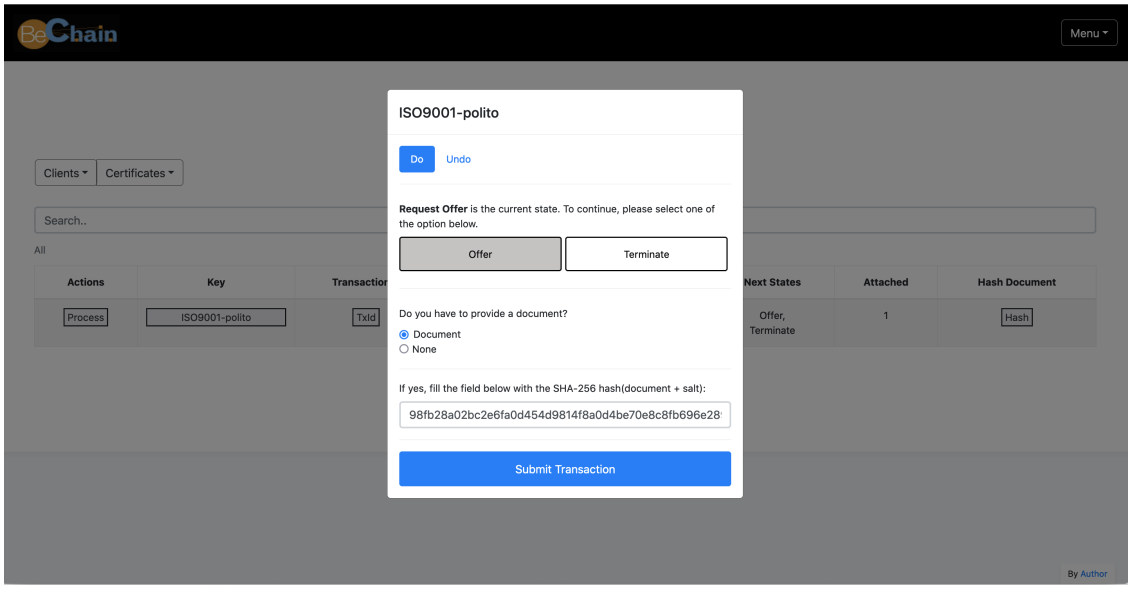

Figura 5.28. Selezioniamo *Document* e digitiamo l'hash più il sale del documento di offerta.

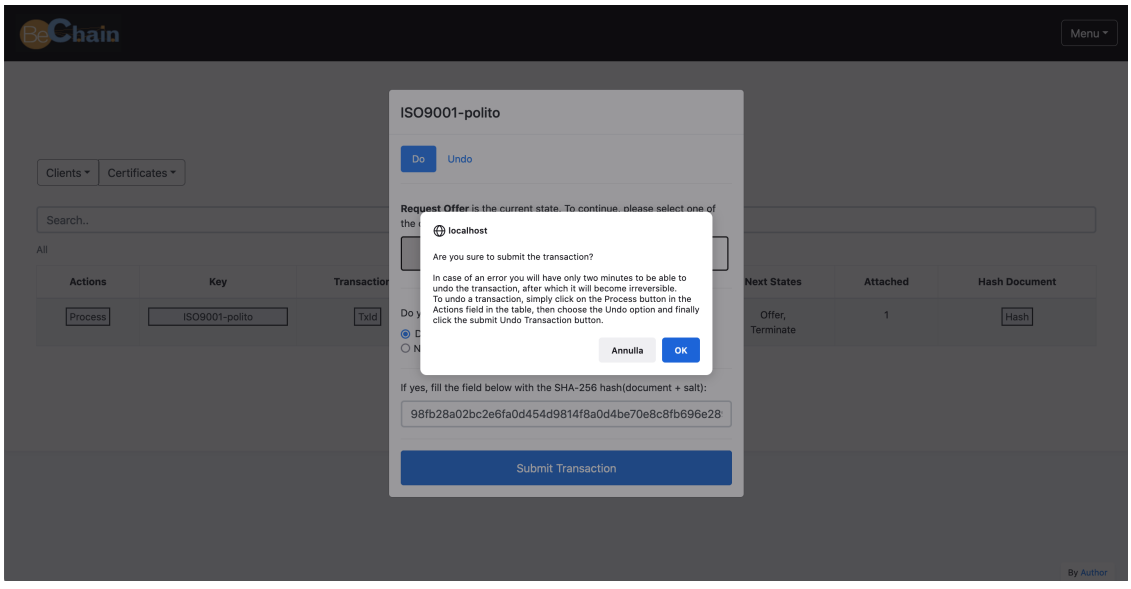

Figura 5.29. Dopo aver cliccato su *Submit Transaction*, viene richiesta un'ulteriore conferma. Clicchiamo su *Ok*.

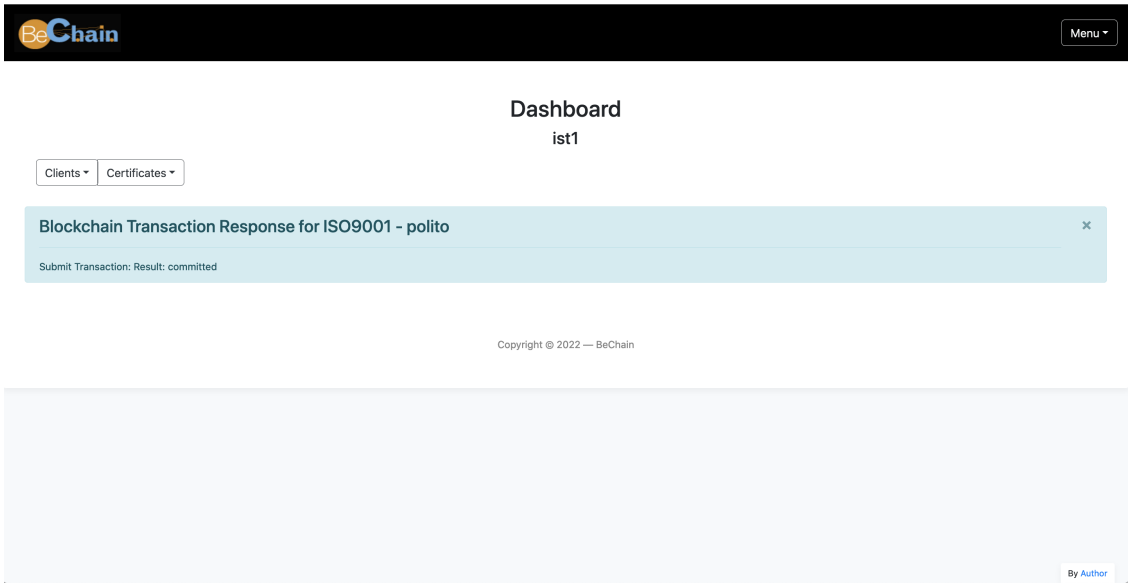

Figura 5.30. Un messaggio di avviso informa che la transazione è stata eseguita con successo.

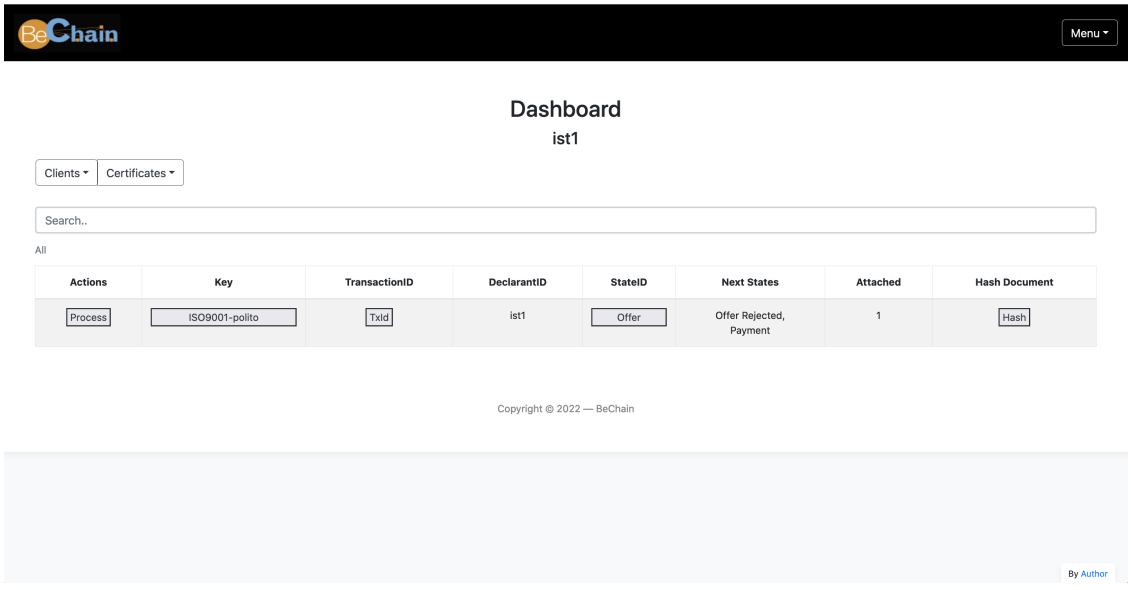

Figura 5.31. Lo stato *ISO9001-polito* è stato aggiornato.

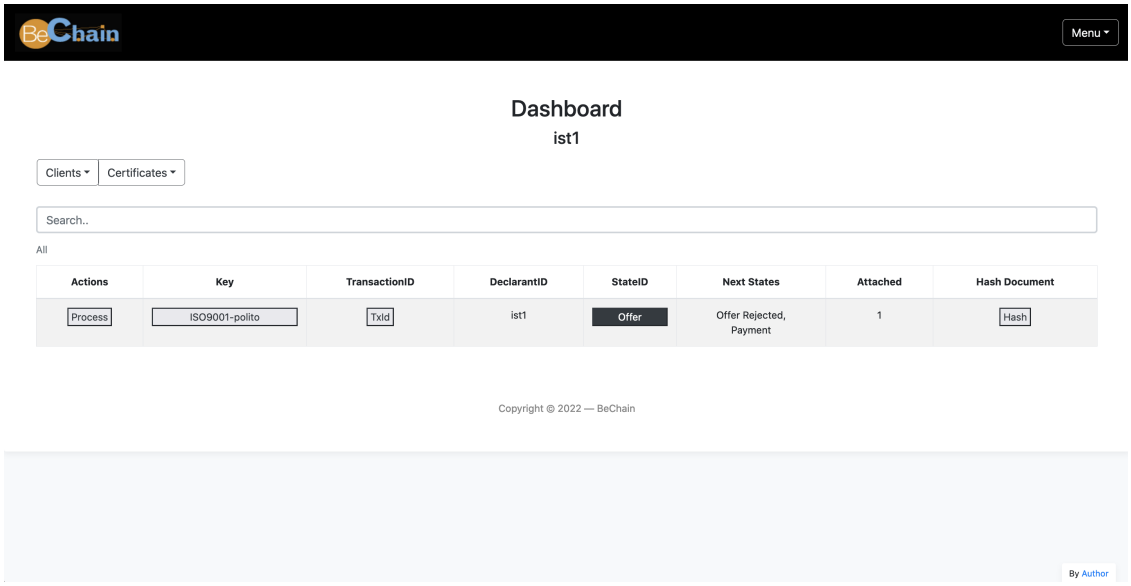

Figura 5.32. Clicchiamo sullo stato corrente *Offer*.

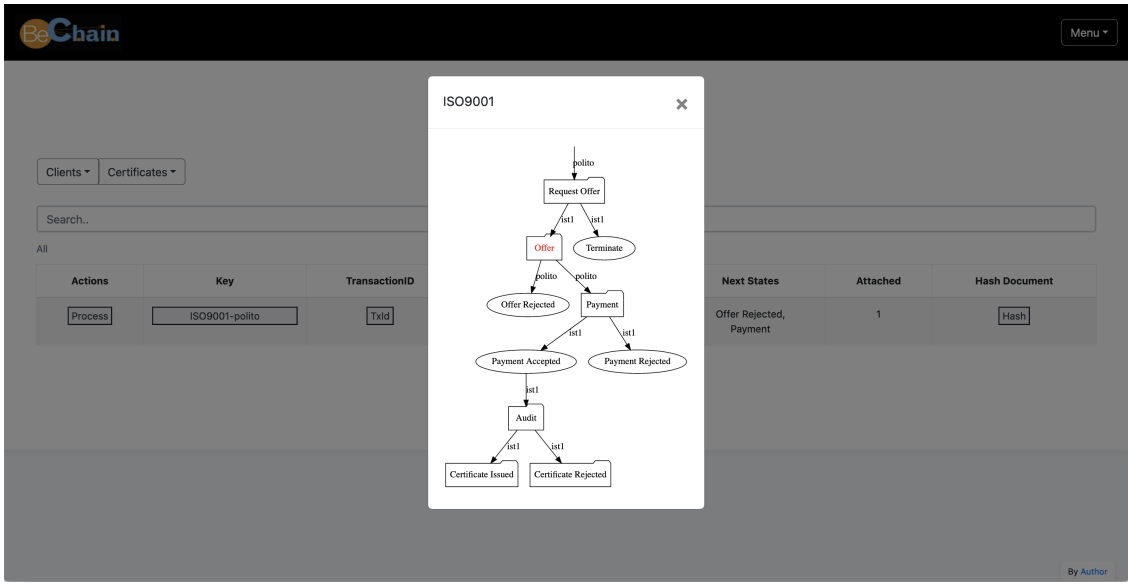

Figura 5.33. Visualizzazione grafica del processo di certificazione. Lo stato corrente è stato aggiornato ad *Offer*.

Annullamento dell'ultima transazione

È possibile annullare una transazione entro due minuti da quando è stata eseguita. Se il tempo è minore di due minuti si assume che l'errore sia di tipo umano.

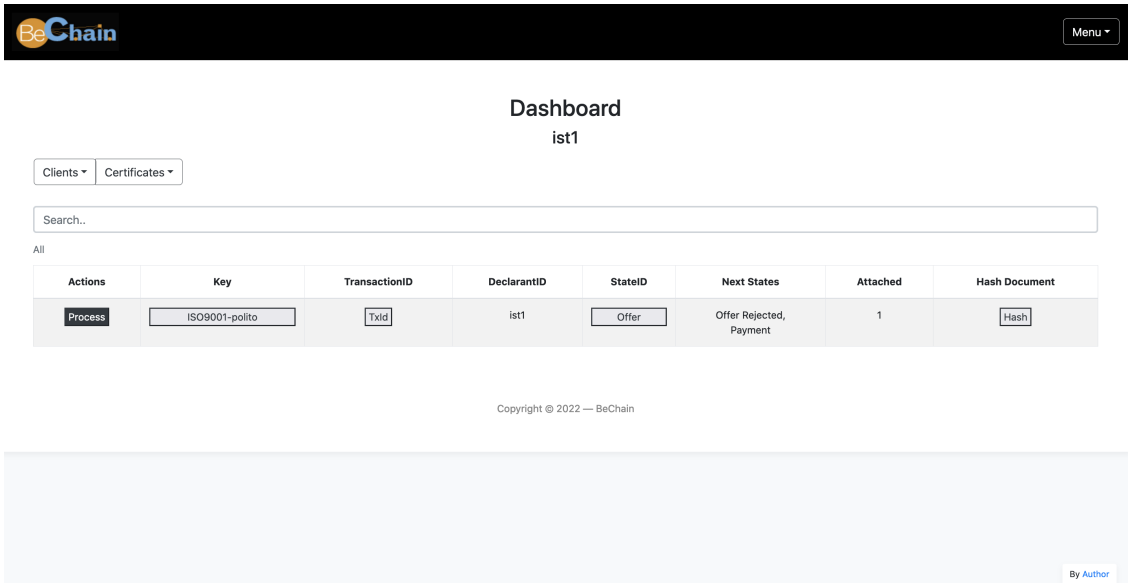

Figura 5.34. Agendo quindi come *ist1* supponiamo di voler annullare la transazione precedente perché abbiamo fornito un hash più il sale non corretto. Clicchiamo su *Process*.

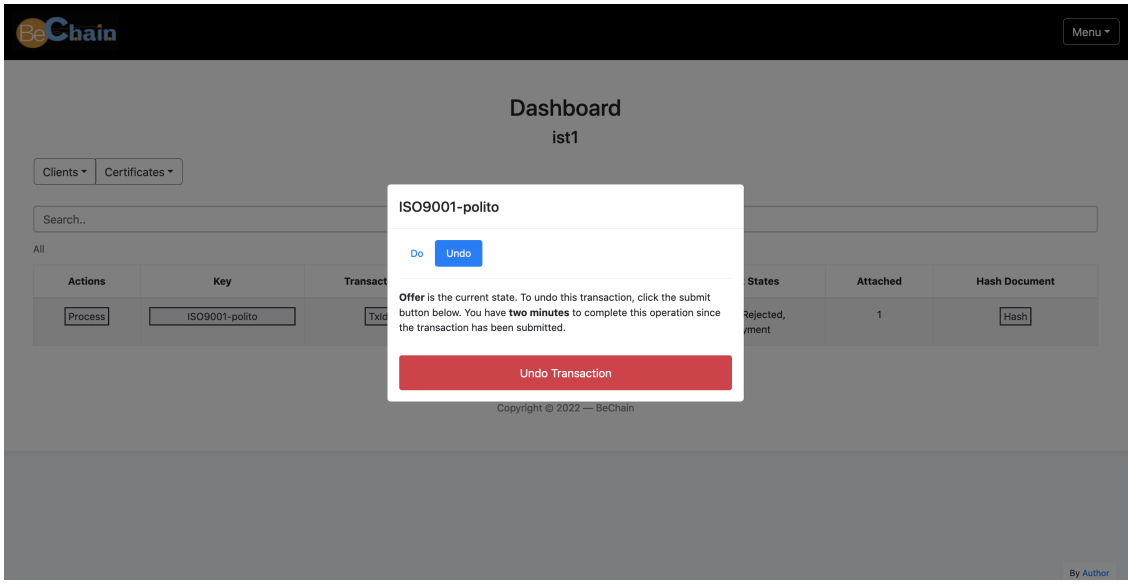

Figura 5.35. Clicchiamo su *Undo* e poi su *Undo Transaction*.

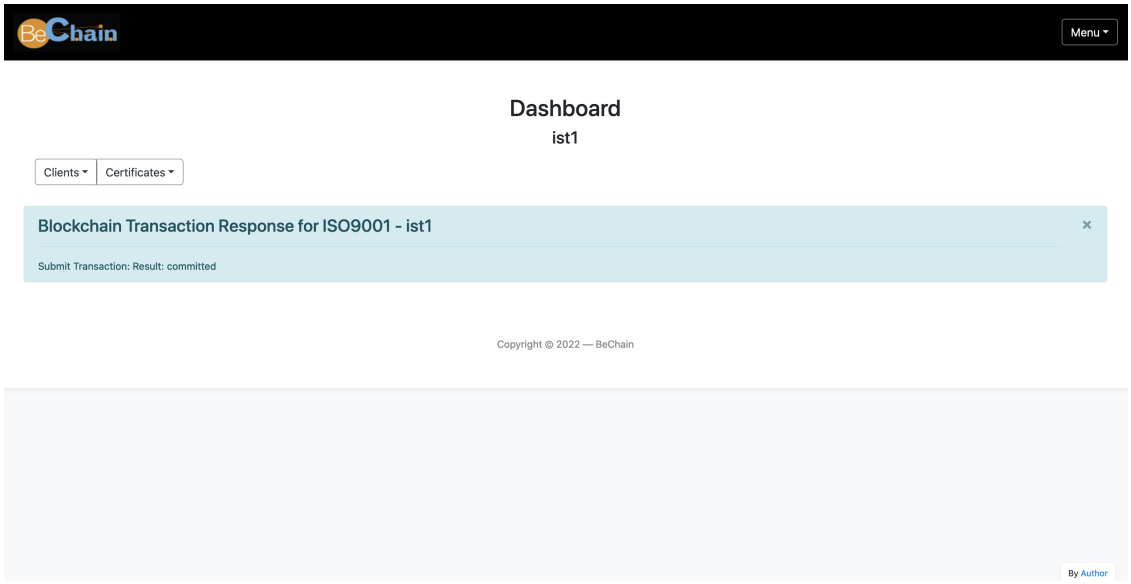

Figura 5.36. Se sono trascorsi meno di due minuti, un messaggio di avviso informa che la transazione è stata eseguita con successo.

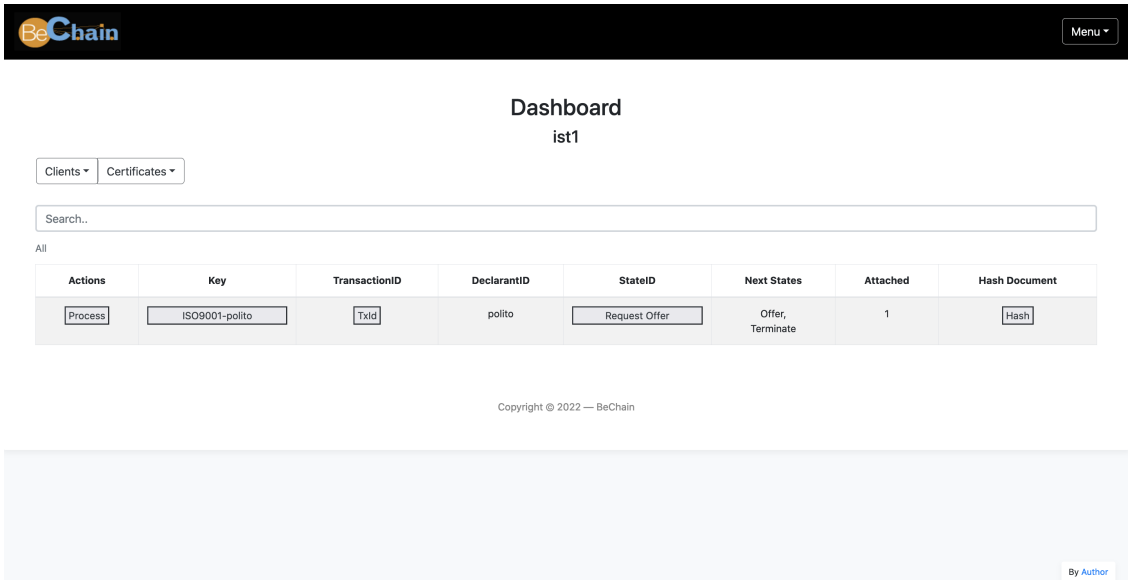

Figura 5.37. Lo stato è stato aggiornato correttamente.

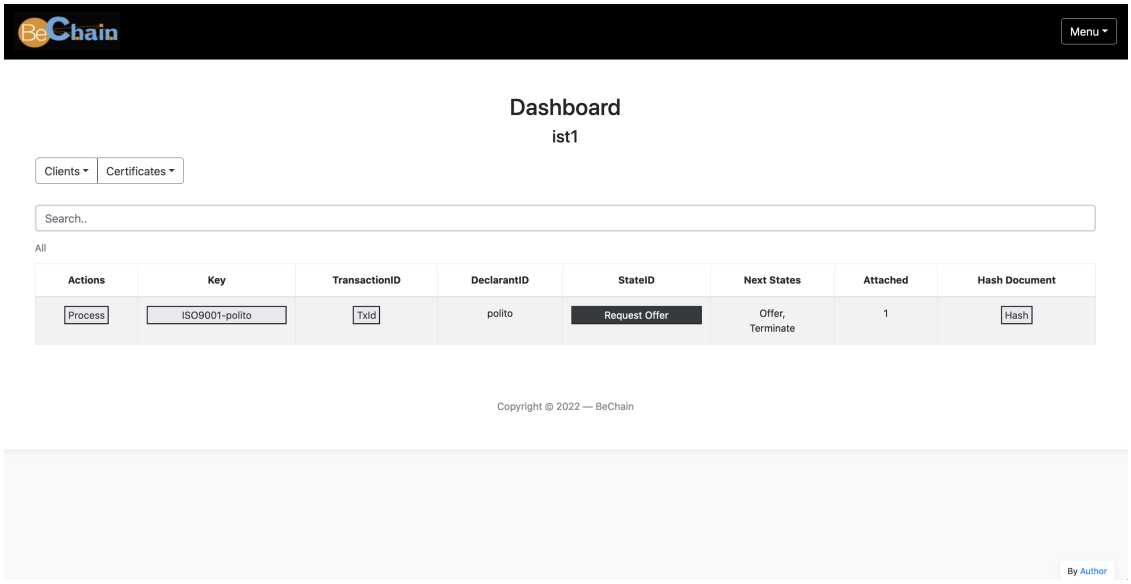

Figura 5.38. Clicchiamo sullo stato corrente *Request Offer*.

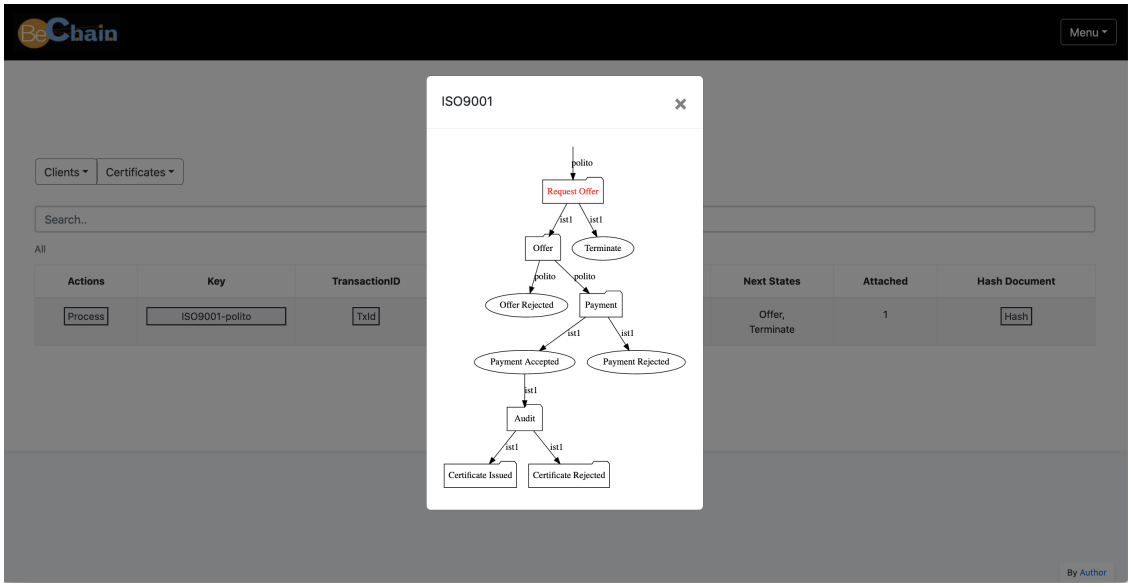

Figura 5.39. Visualizzazione grafica del processo di certificazione. Lo stato corrente è *Request Offer*.

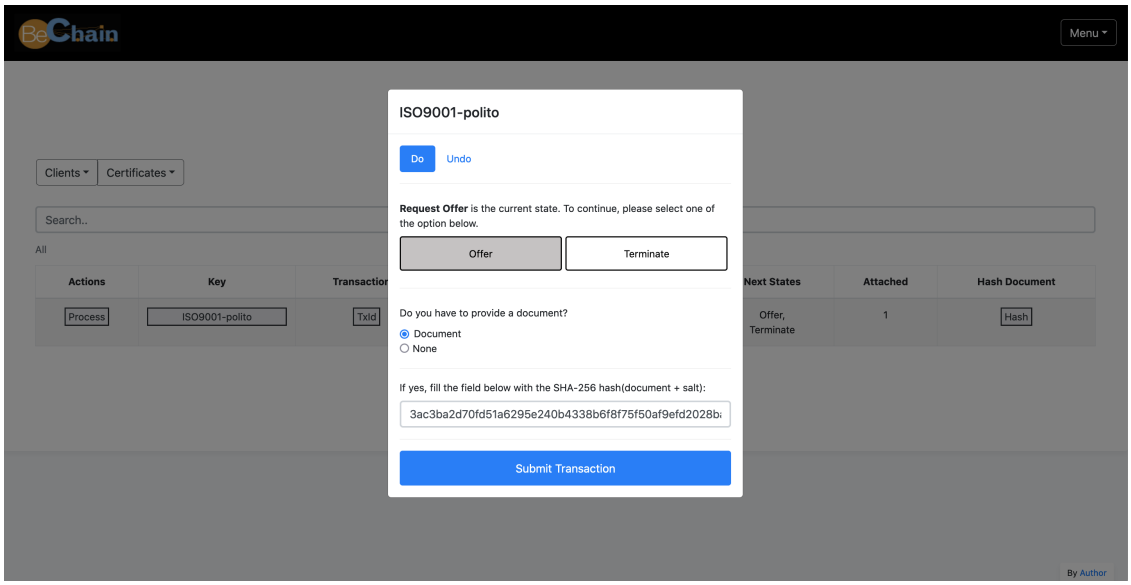

Figura 5.40. Eseguiamo nuovamente la transazione digitando ora l'hash più il sale corretto.

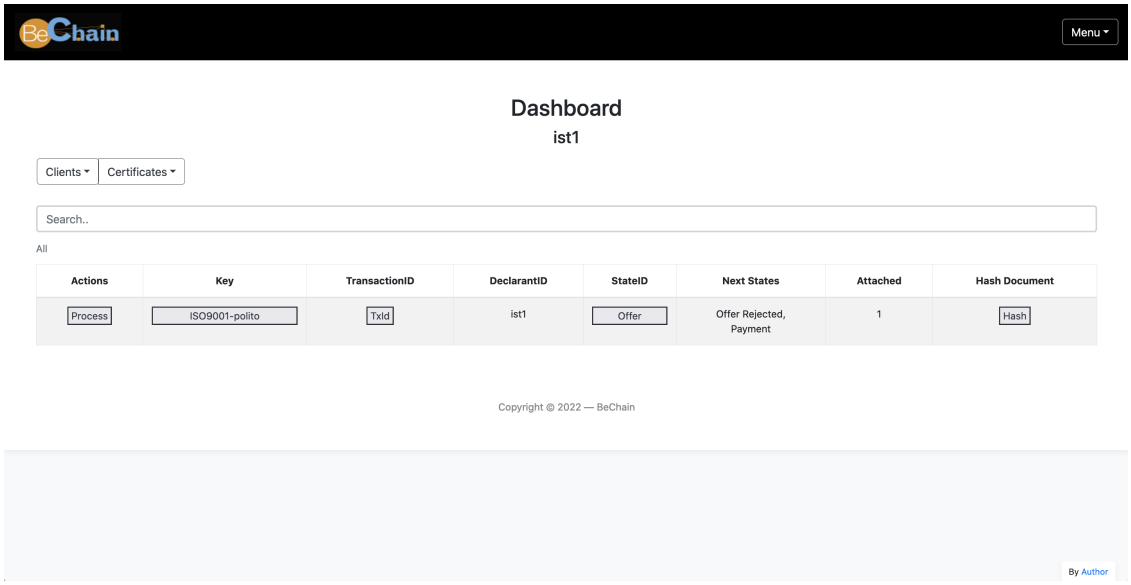

Figura 5.41. Visualizzazione dello stato mondiale.

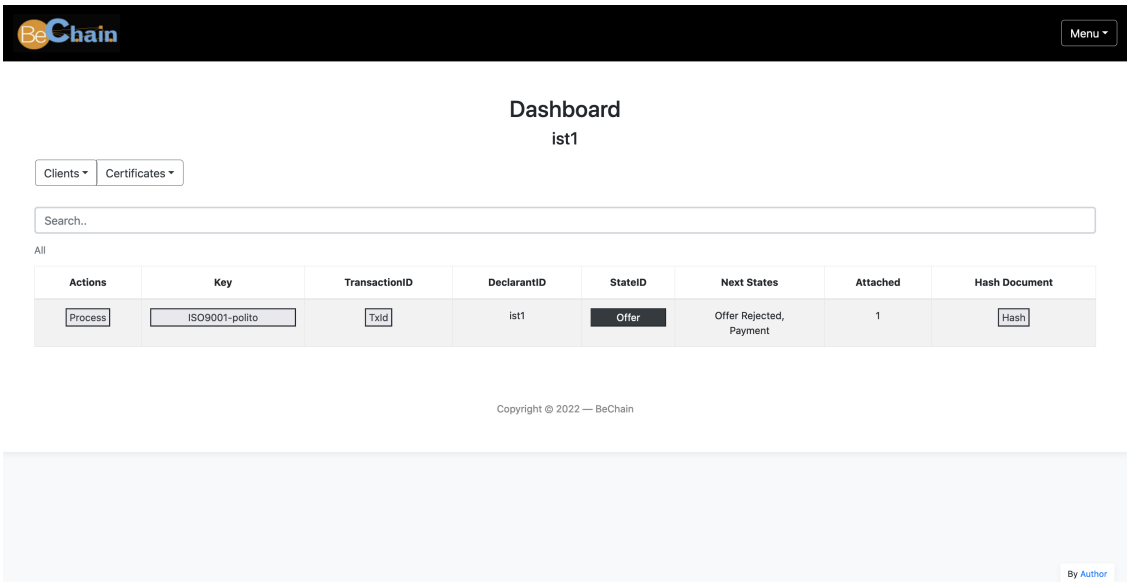

Figura 5.42. Clicchiamo sullo stato corrente di *Offer*.

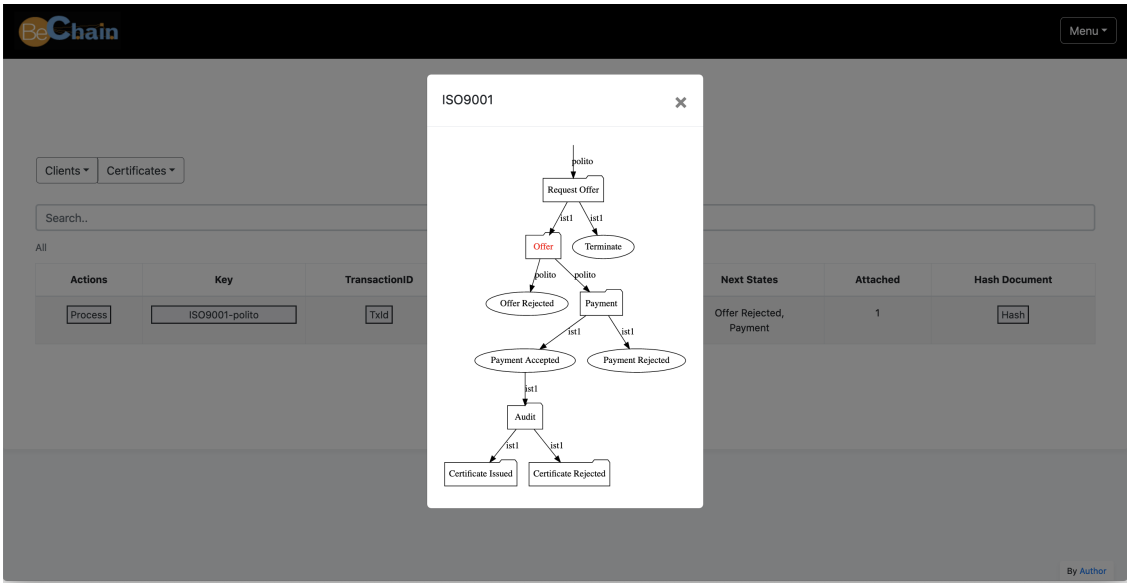

Figura 5.43. Visualizzazione grafica del processo di certificazione. Lo stato corrente è Offer.

Visualizzazione dei prossimi stati nel processo di certificazione

È possibile visualizzare i prossimi stati del processo di certificazione in corrispondenza della colonna *Next States*.

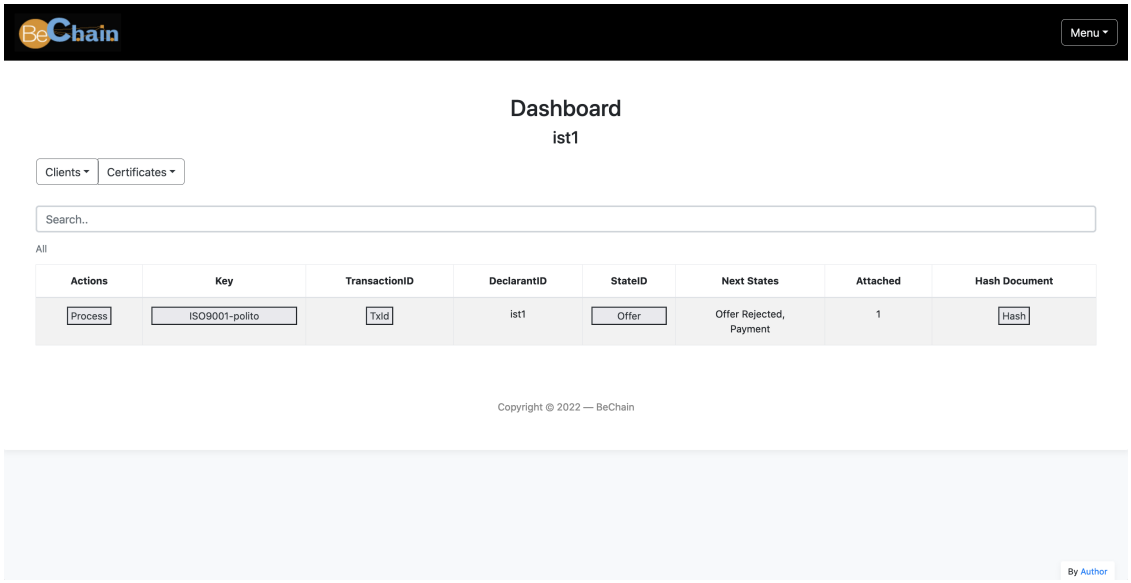

Figura 5.44. Visualizzazione dello stato mondiale.

Visualizzazione delle transazioni precedenti

È possibile visualizzare tutte le transazioni passate registrate nella blockchain relativamente a uno dei processi di certificazione visibili nello stato mondiale.

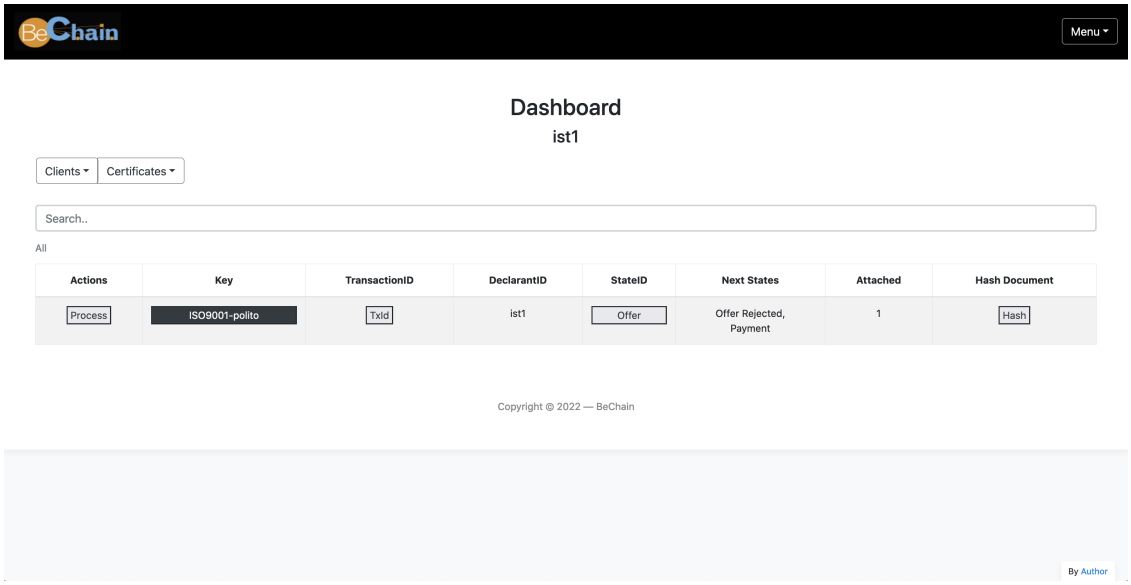

Figura 5.45. Dalla pagina dell'*ist1*, clicchiamo sulla chiave *ISO9001-polito*.

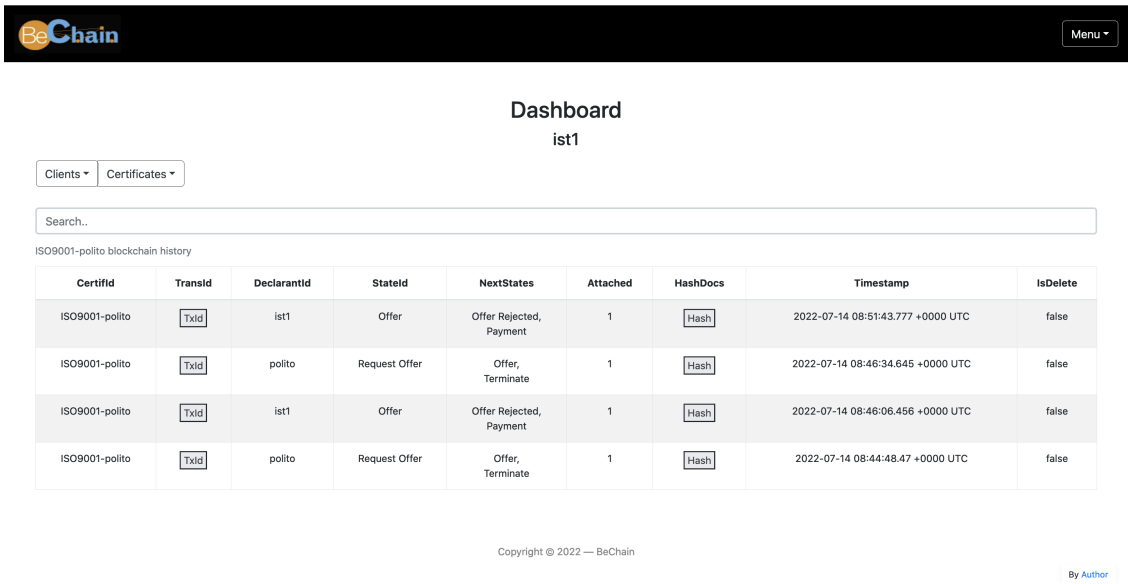

Figura 5.46. Visualizzazione di tutte le transazioni avvenute tra l'*ist1* e *polito* in merito al certificato ISO9001. L'ordine cronologico è dal basso verso l'alto. Si noti in particolare alla riga due e tre l'annullamento della transazione da parte dell'*ist1*.

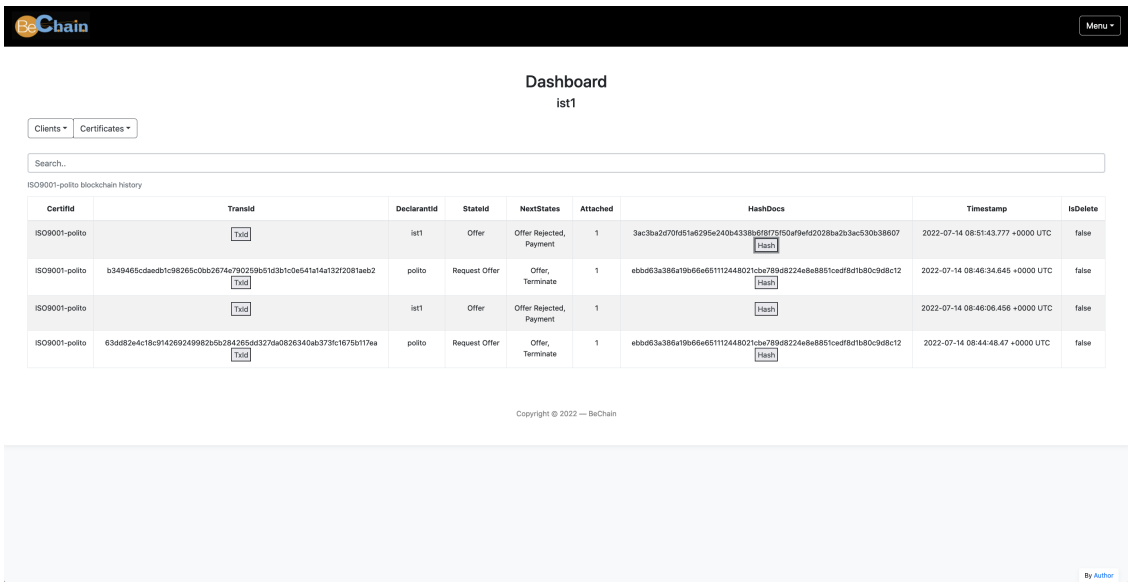

Figura 5.47. In merito alla transazione annullata, essendo tornati indietro nella macchina a stati, notiamo che l'hash del documento è sempre lo stesso. Si tratta però di transazioni diverse come indicato dai differenti id delle transazioni.

Inserimento del codice d'invito di un'altra istituzione

In questo paragrafo vediamo come un cliente già registrato diventa cliente di un'altra istituzione, l'*ist2*.

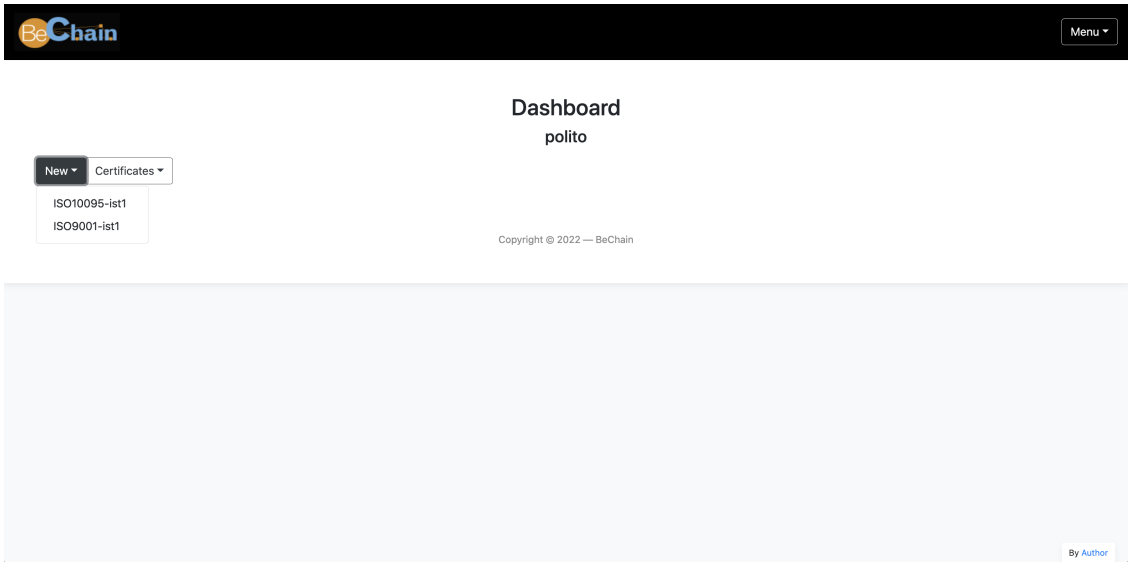

Figura 5.48. Al momento il cliente *polito* può avviare un processo di certificazione solo con l'*ist1* e relativamente ai certificati ISO9001 e ISO10095 da lui offerti.

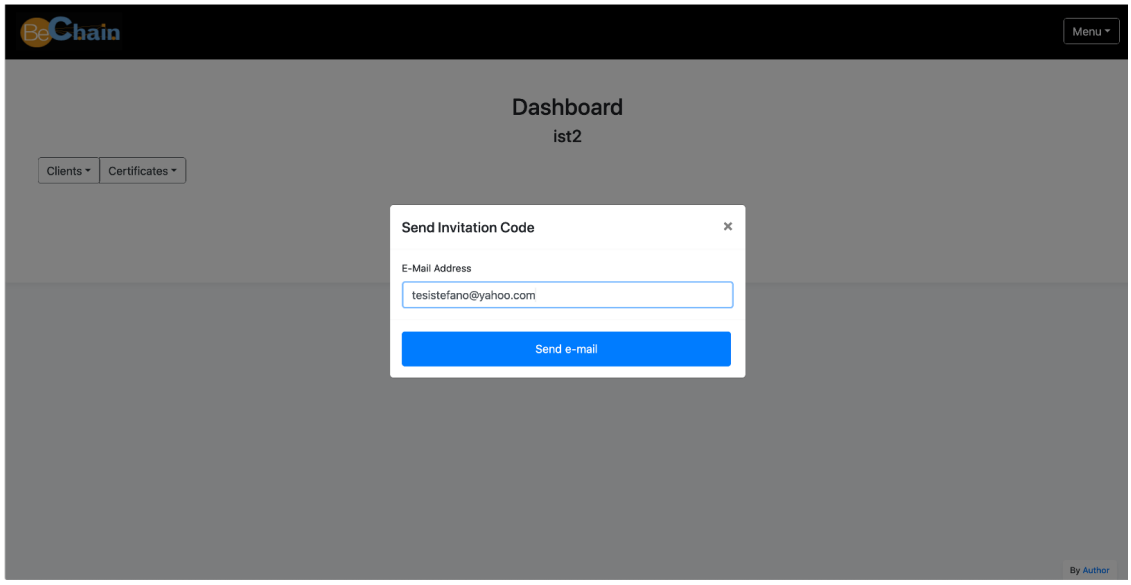

Figura 5.49. l'*ist2* invia un codice di invito all'utente *polito*.

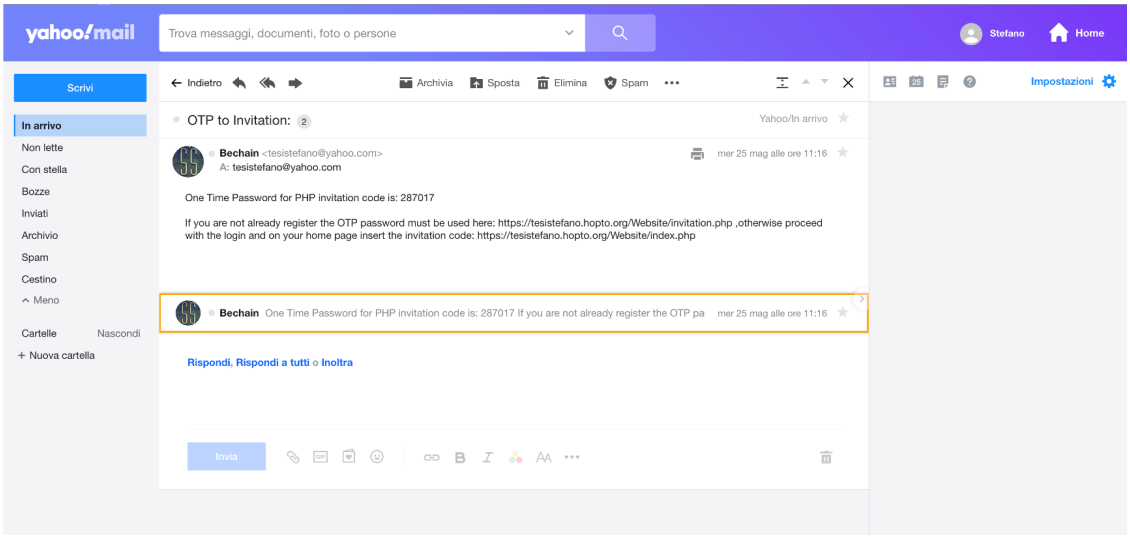

Figura 5.50. *Polito* riceve il codice di invito nella sua casella di posta.

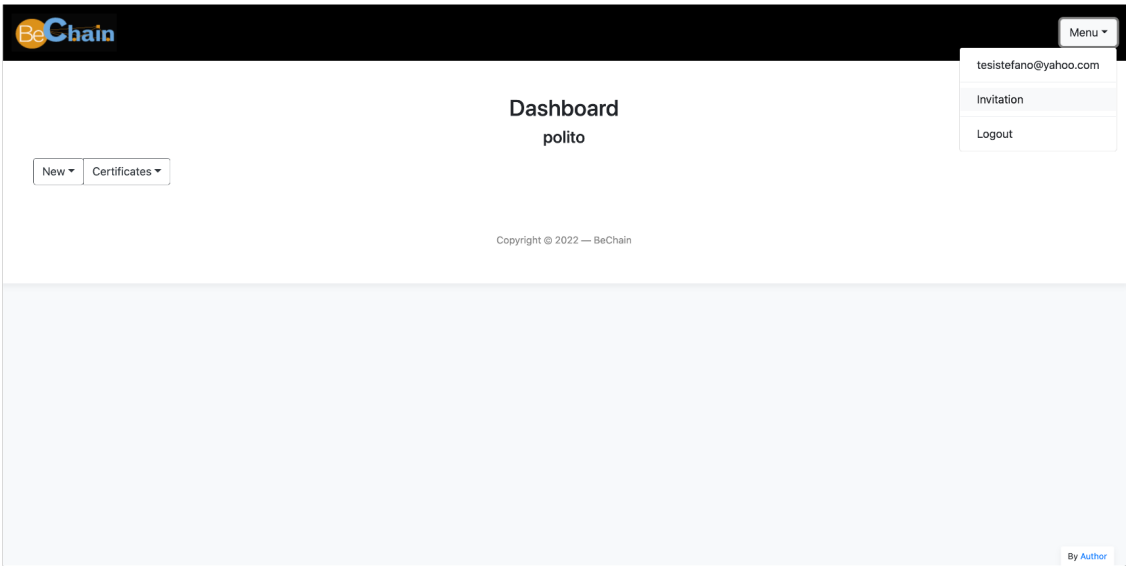

Figura 5.51. Dalla sua pagina personale *polito* clicca su *Menu* e successivamente su *Invitation*.

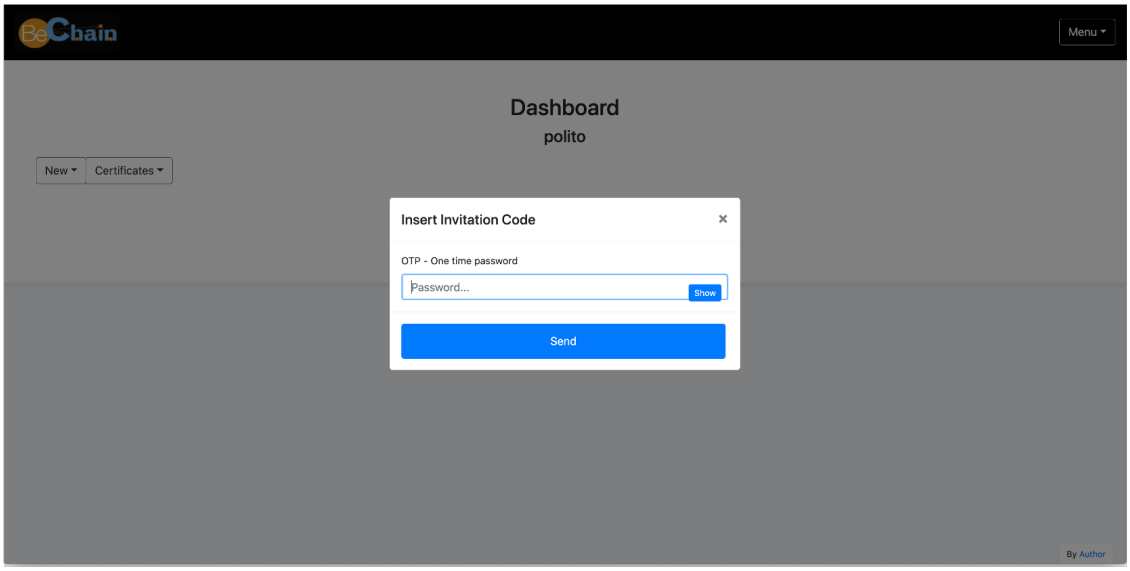

Figura 5.52. Form di inserimento del codice di invito.

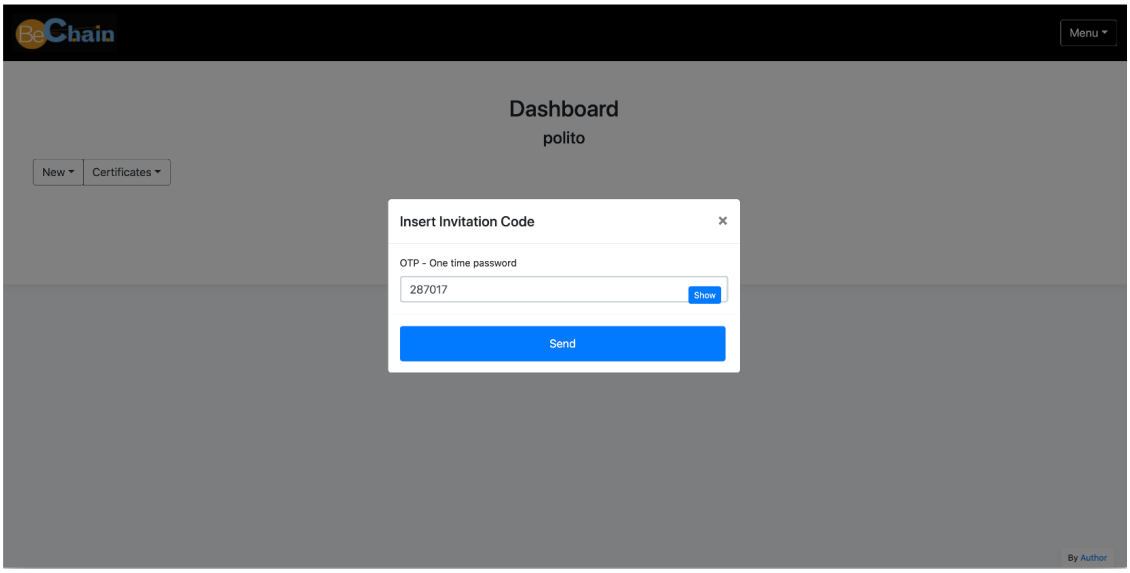

Figura 5.53. Digitiamo nel campo del form il codice di invito ricevuto per email e successivamente clicchiamo su *Send*.

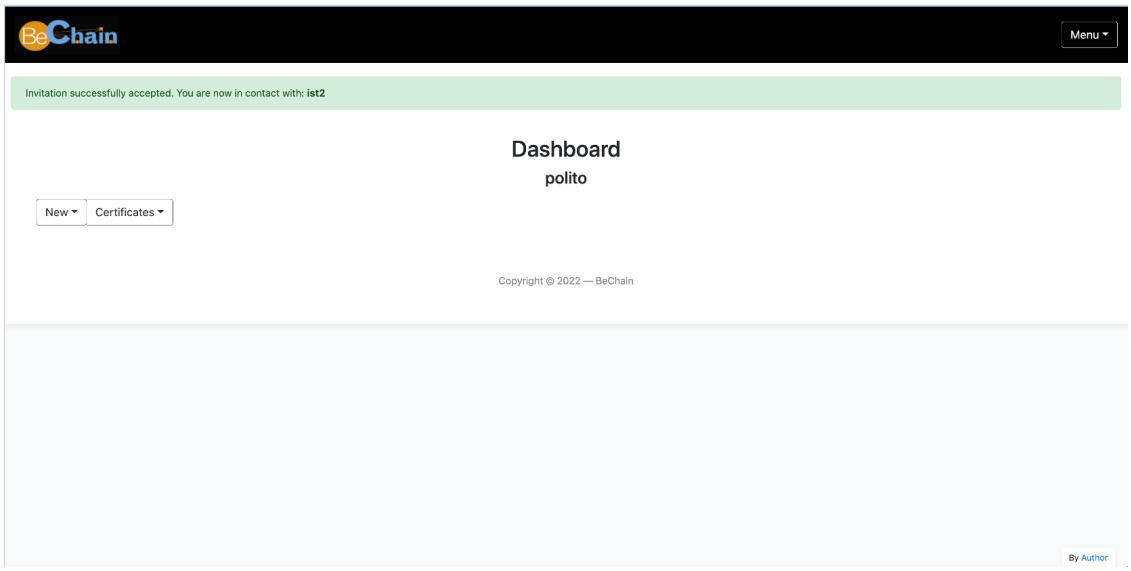

Figura 5.54. Messaggio di avviso: codice inserito corretto.

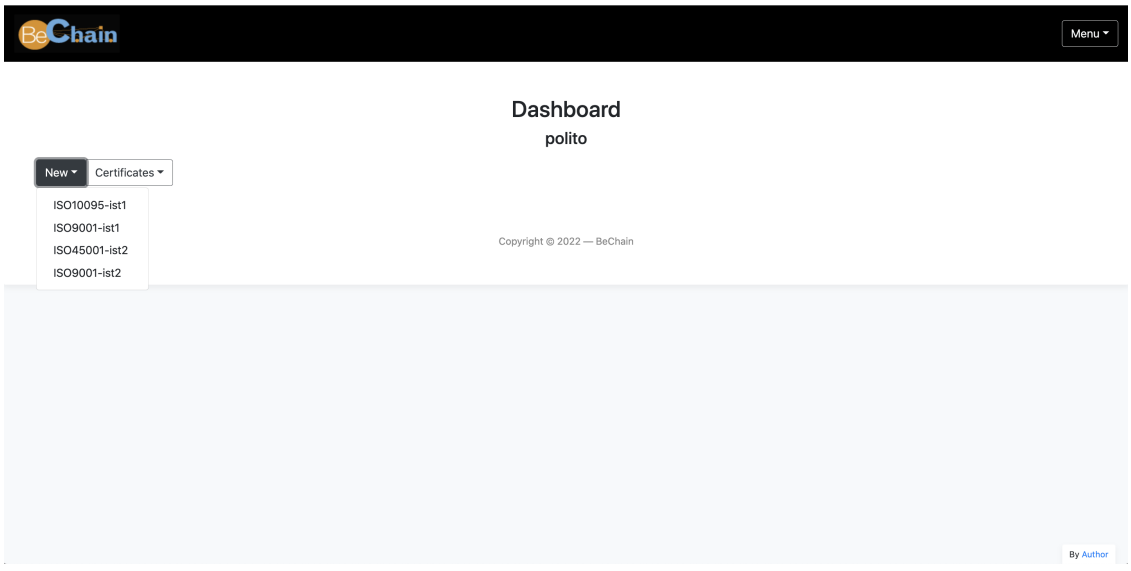

Figura 5.55. *Polito* può ora avviare un processo di certificazione anche con l'ist2 relativamente ai certificati offerta dall'istituzione. In particolare i certificati sono l'ISO45001 e l'ISO9001.

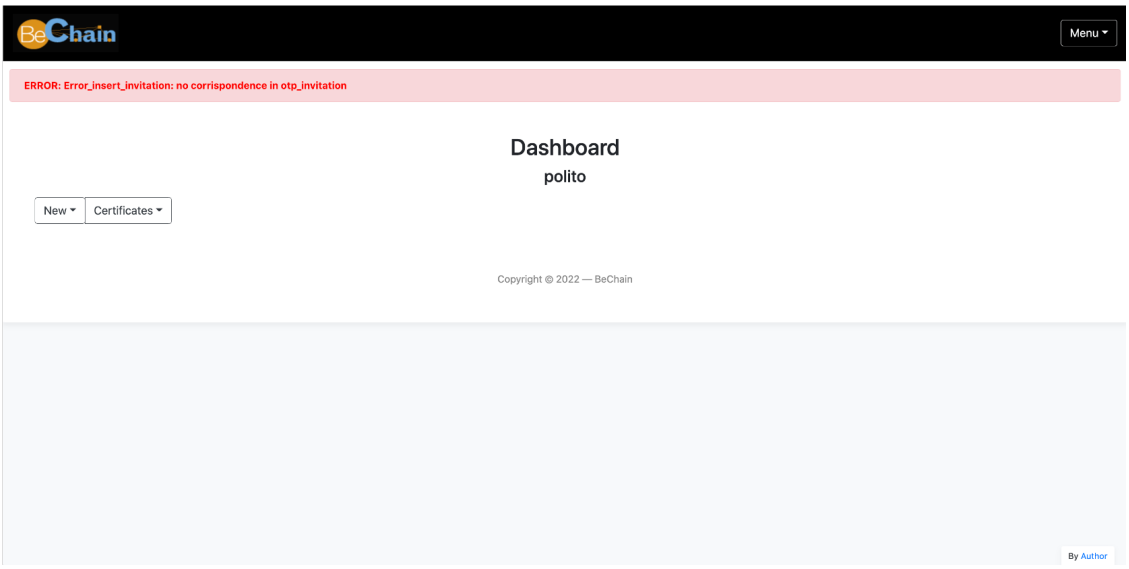

Figura 5.56. Se inseriamo un codice di invito non valido appare un messaggio di errore.

Inizializzazione di un nuovo processo di certificazione

In questo paragrafo il nuovo cliente *unito* avvia un processo di certificazione con l'*ist1* relativamente al certificato ISO10095.

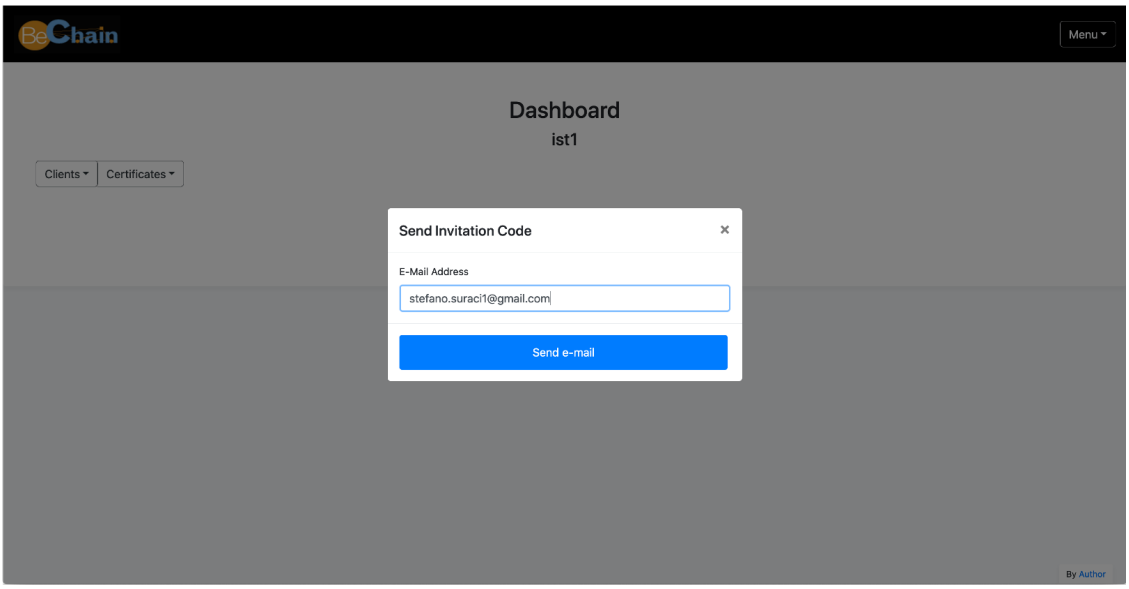

Figura 5.57. L'*ist1* invia il codice di invito a *unito*.

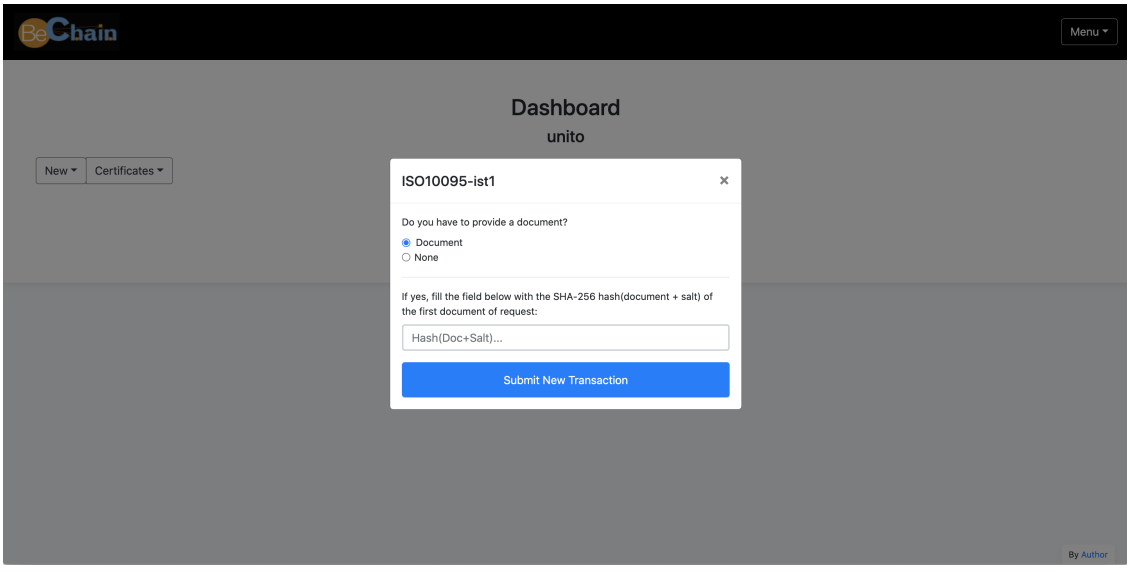

Figura 5.58. Dopo aver accettato l'invito, *unito* richiede di essere certificato per l'ISO10095.

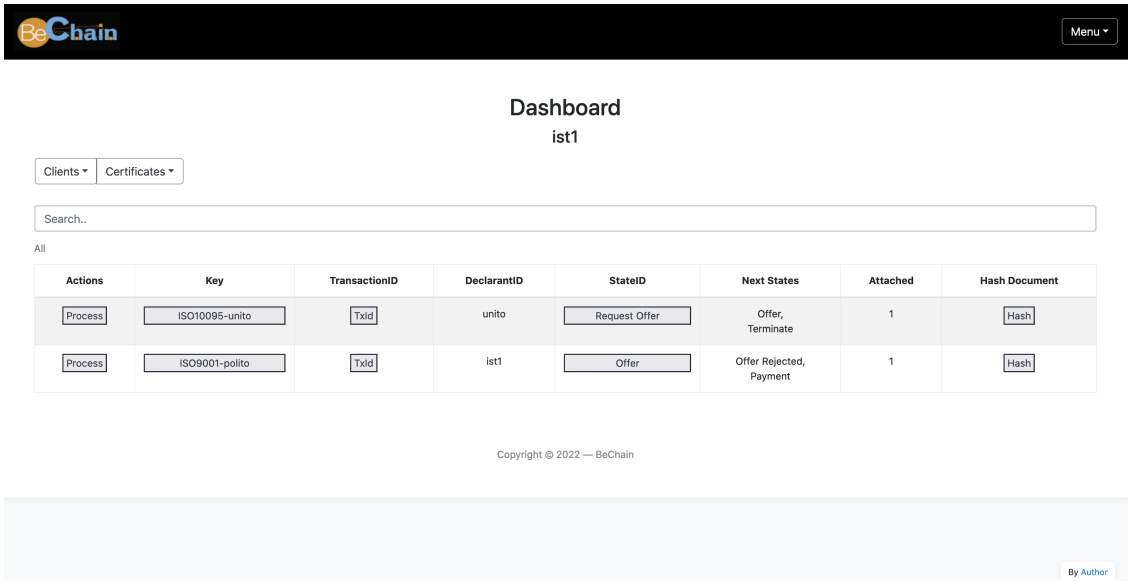

Figura 5.59. Nella pagina personale dell'*ist1* sono ora visibili entrambe le ultime transazioni effettuate con *polito* e *unito*.

Visualizzazione di tutti i processi di certificazione dal punto di vista di una istituzione

In questo paragrafo sono indicati i vari passaggi per poter visualizzare tutti i processi di certificazione presenti nello stato mondiale di un'istituzione.

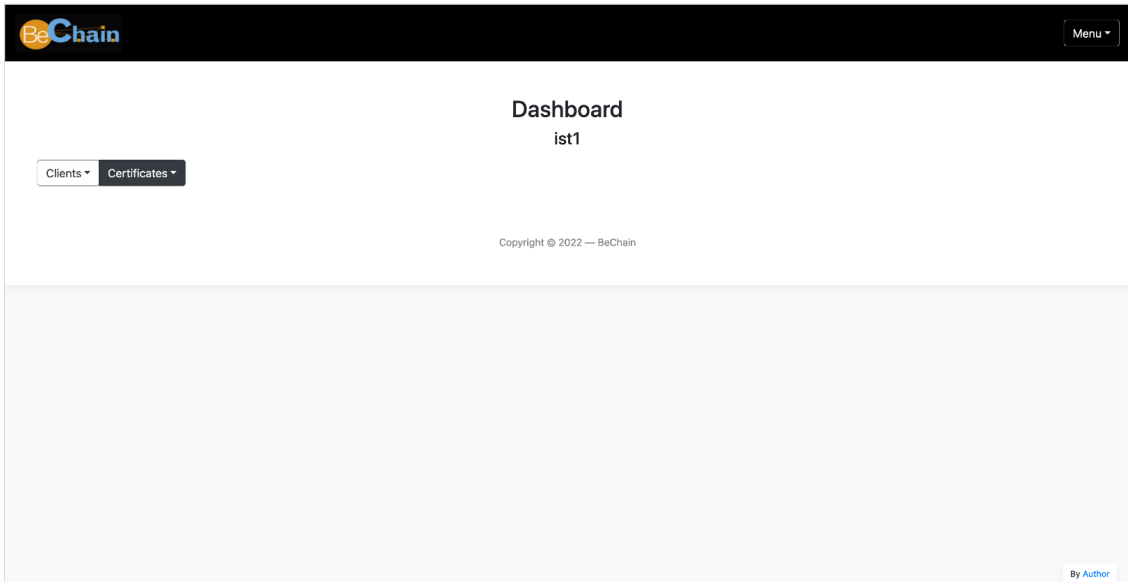

Figura 5.60. Dalla pagina personale clicchiamo su *Certificates*.

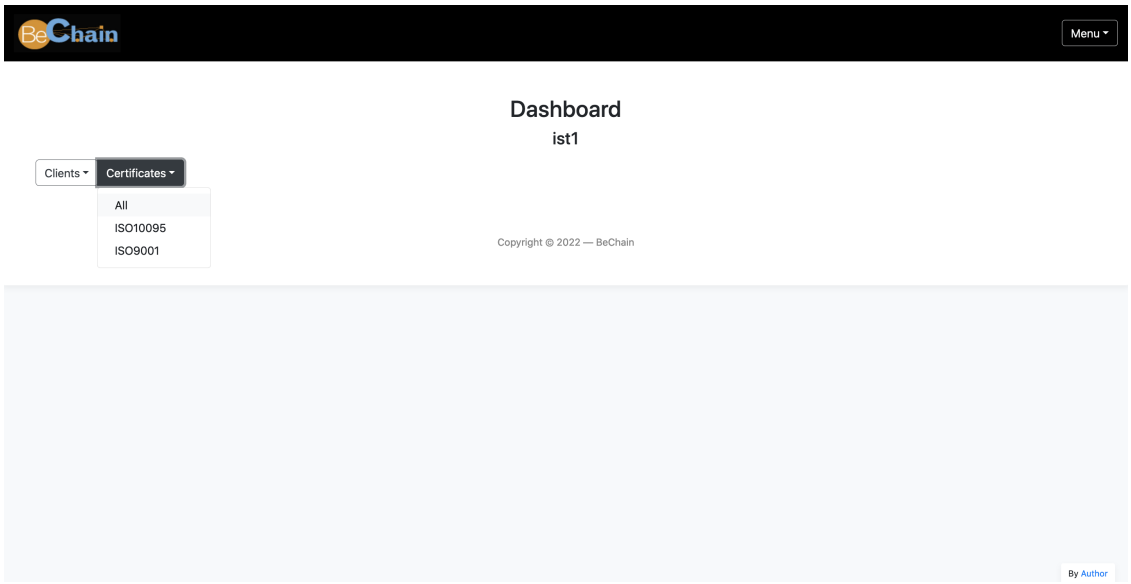

Figura 5.61. Clicchiamo su *All*.

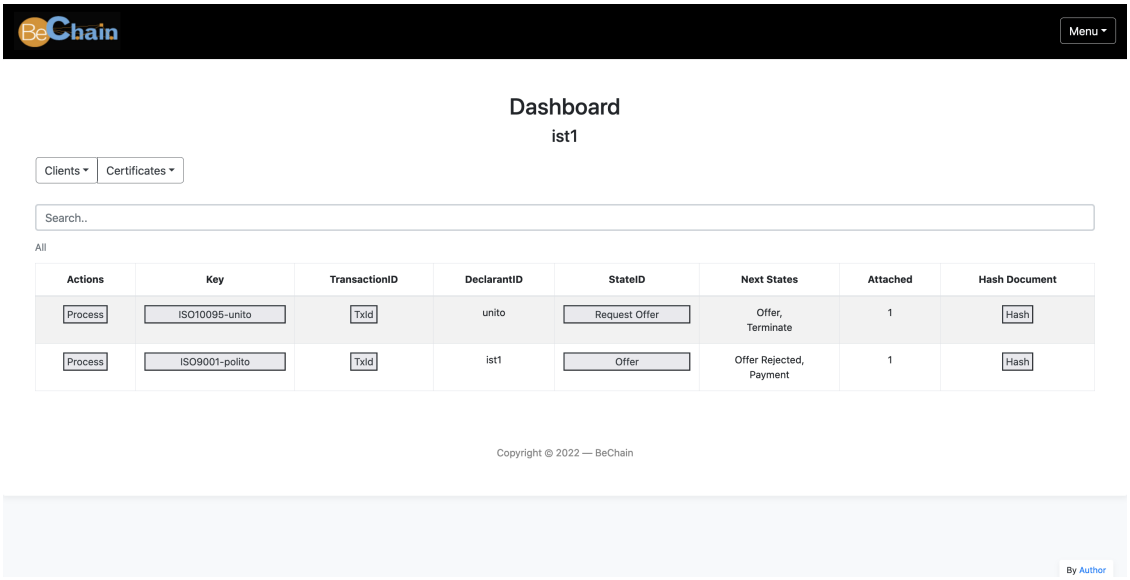

Figura 5.62. Visualizzazione di tutte le ultime transazioni effettuate tra l'*ist1* e ognuno dei suoi clienti.

Visualizzazione di tutti i processi di certificazione ISO10095 e ISO9001 agendo come istituzione

In questo paragrafo sono indicati i vari passaggi per poter visualizzare agendo come istituzione tutti i processi di certificazione relativi al contratto ISO10095 e ISO9001.

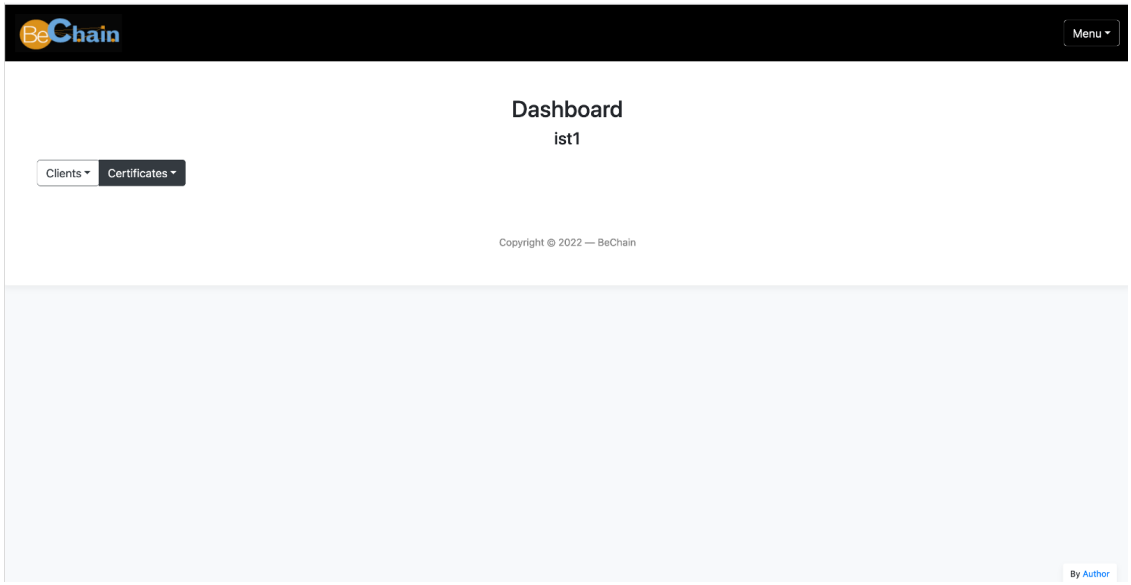

Figura 5.63. Dalla pagina personale clicchiamo su *Certificates*.

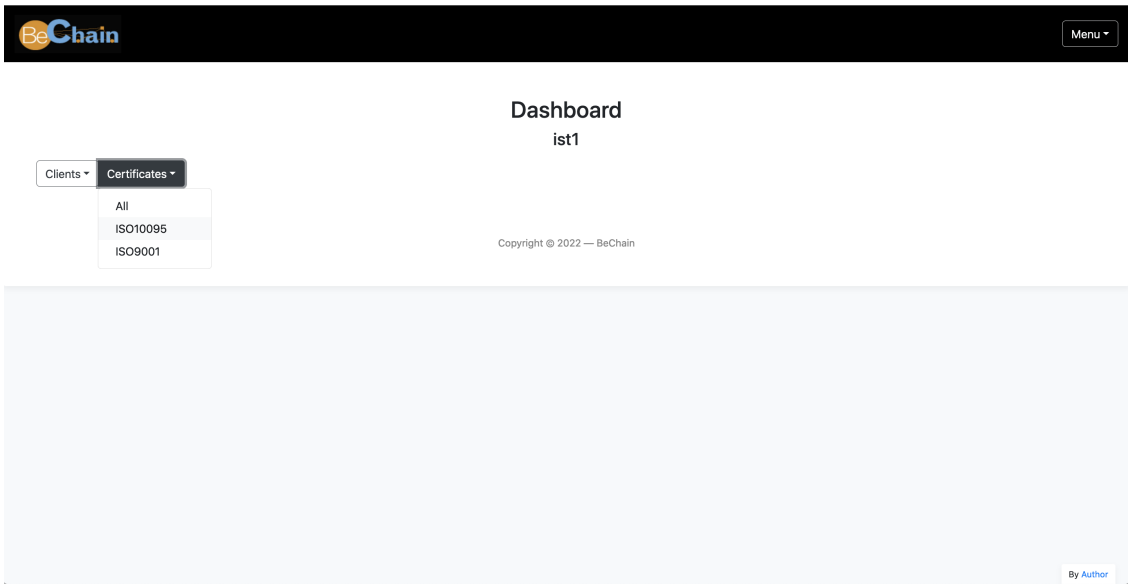

Figura 5.64. Clicchiamo su *ISO10095*.

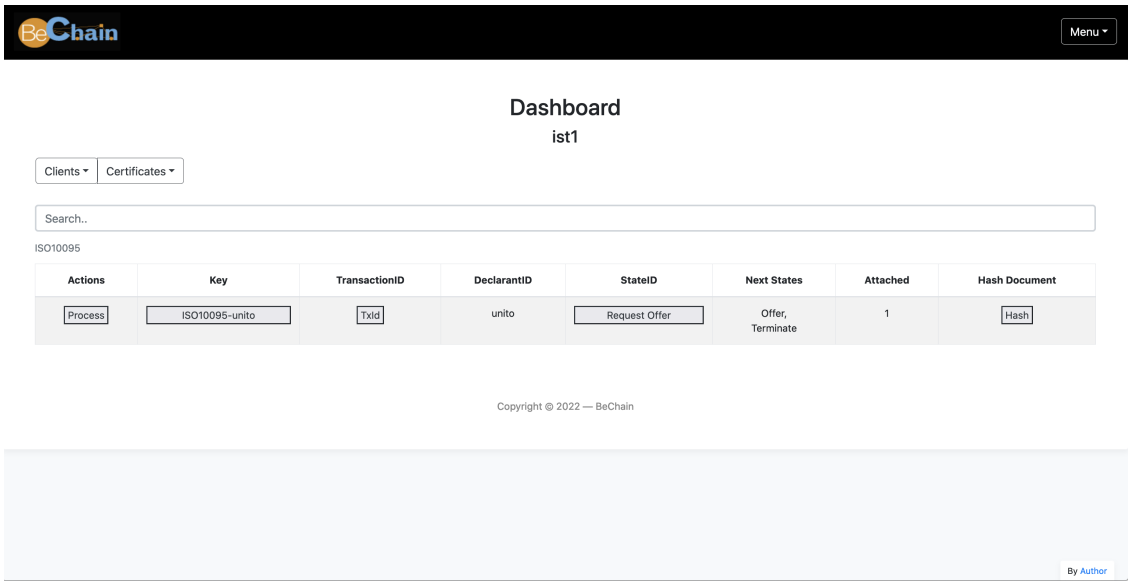

Figura 5.65. Visualizzazione di tutti i processi di certificazione relativi al contratto ISO10095.

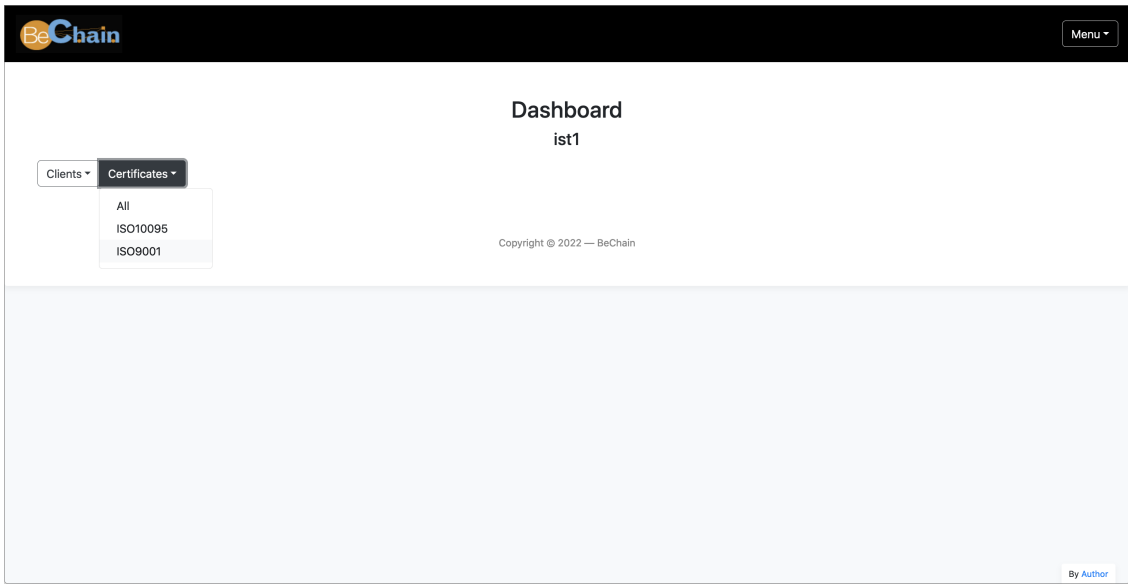

Figura 5.66. Clicchiamo su *ISO9001*.

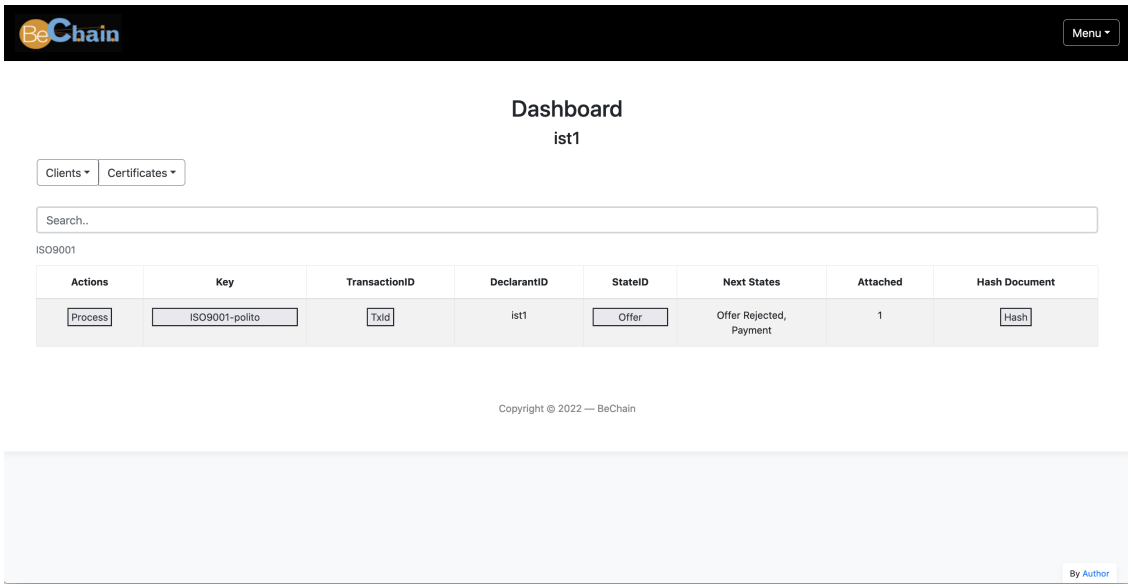

Figura 5.67. Visualizzazione di tutti i processi di certificazione relativi al contratto ISO9001.

Visualizzazione di tutti i processi di certificazione con un determinato cliente

In questo paragrafo sono indicati i vari passaggi per poter visualizzare tutti i processi di certificazione relativi a un determinato cliente.

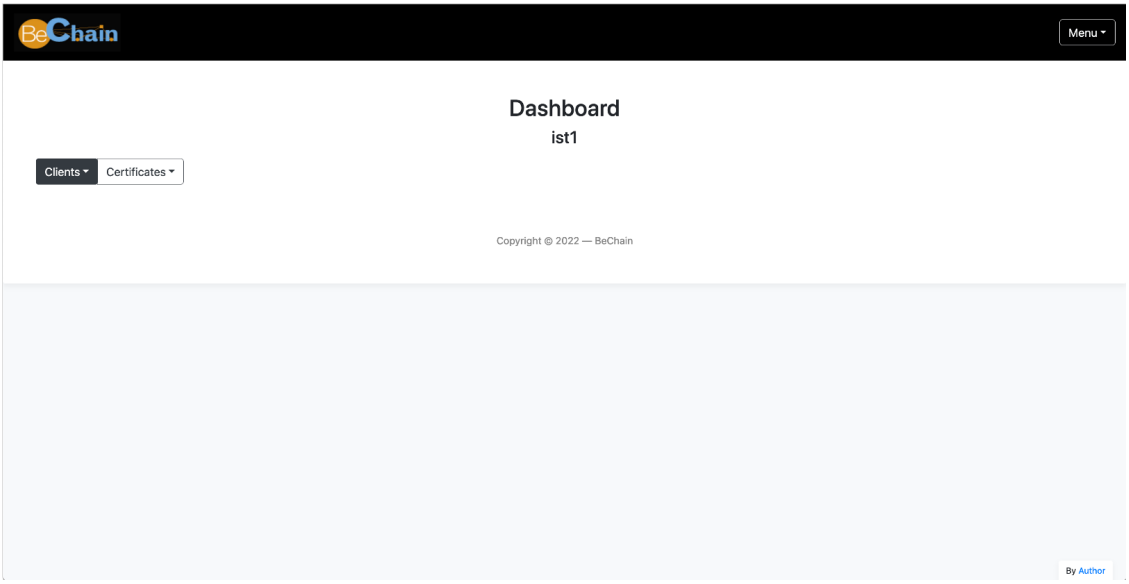

Figura 5.68. Dalla pagina personale clicchiamo su *Clients*.

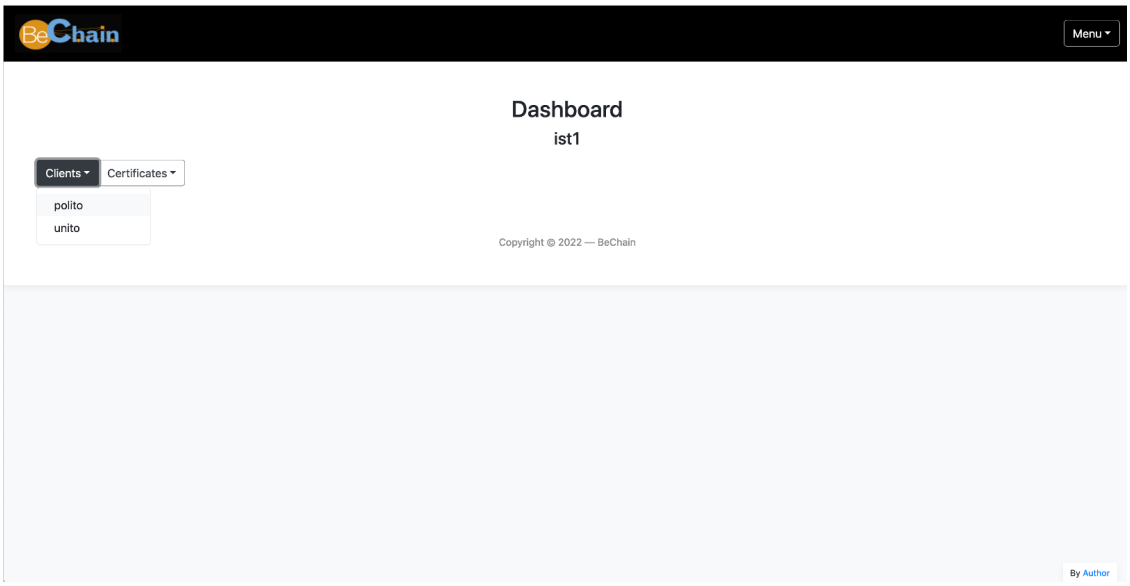

Figura 5.69. Clicchiamo su *polito*.

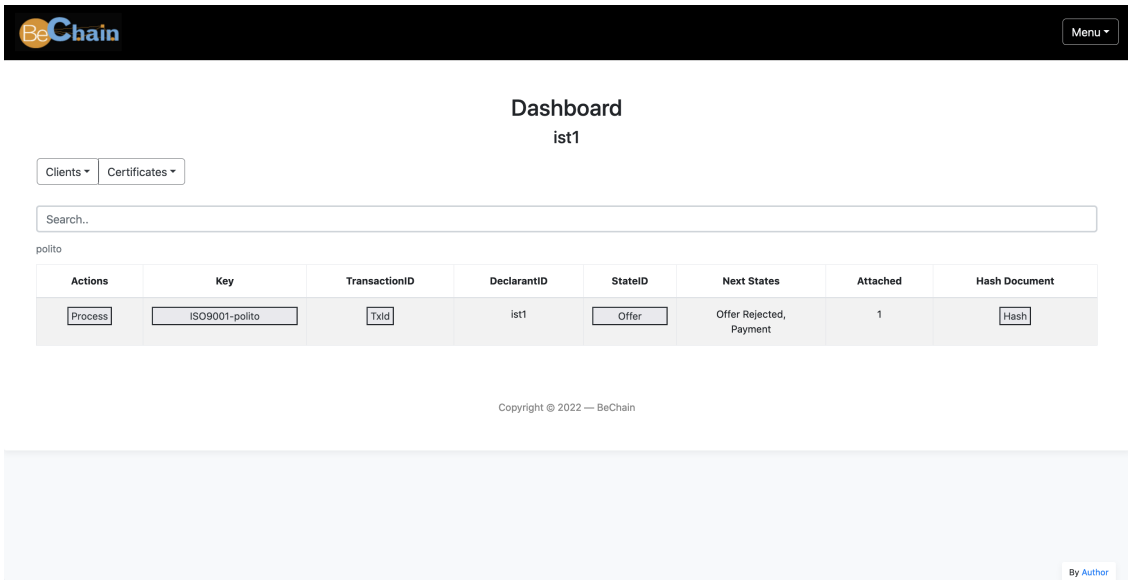

Figura 5.70. Visualizzazione di tutti i processi di certificazione relativi al cliente *polito*.

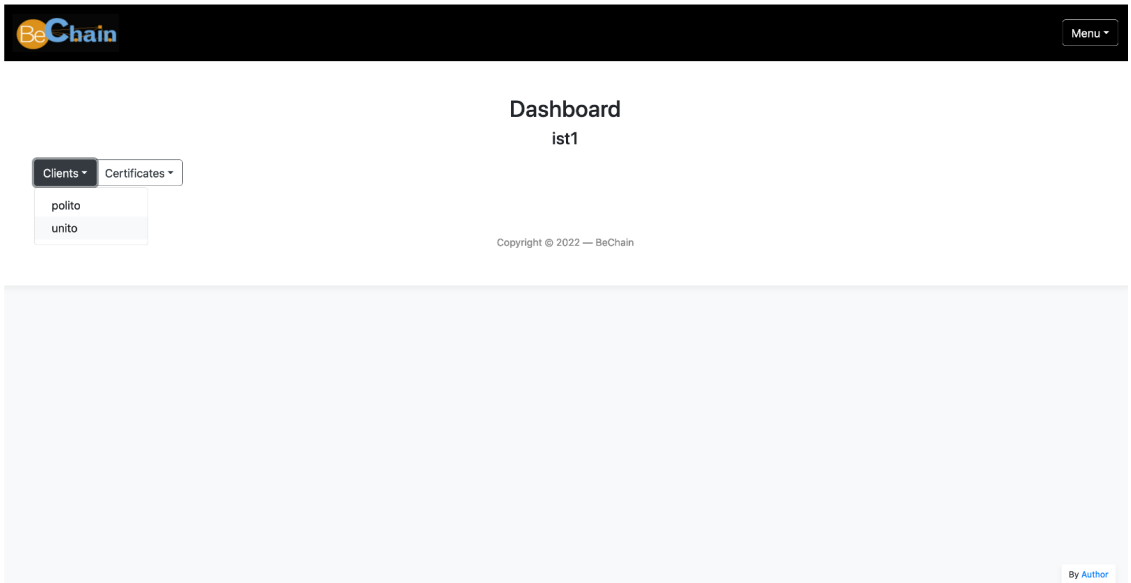

Figura 5.71. Clicchiamo su *unito*.

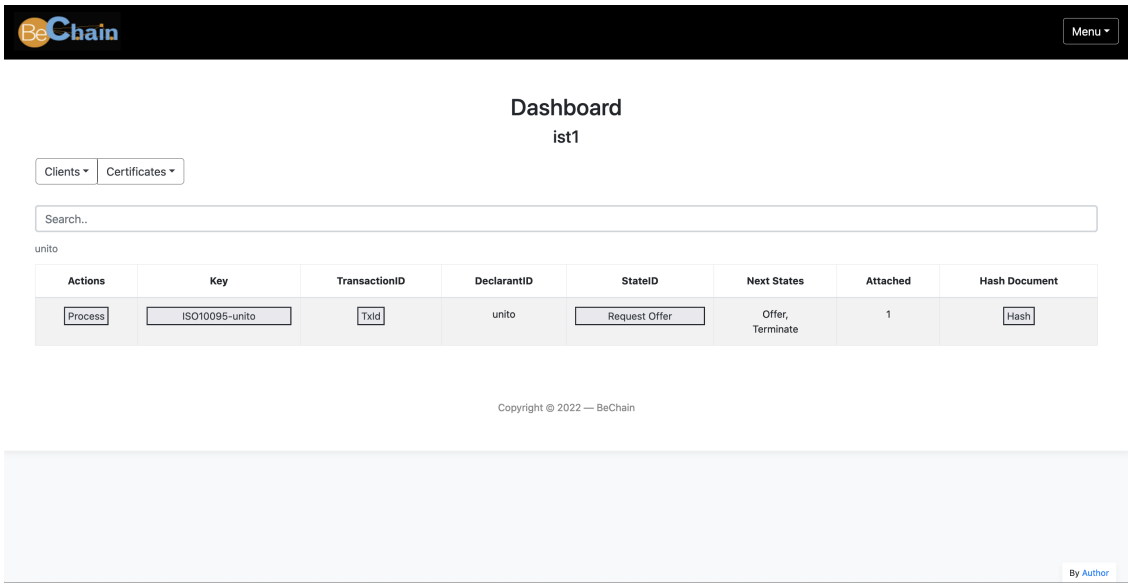

Figura 5.72. Visualizzazione di tutti i processi di certificazione relativi al cliente *unito*.

Visualizzazione di tutti i processi di certificazione combinando diverse chiavi di ricerca

In questo paragrafo sono indicati i vari passaggi per poter visualizzare tutti i certificati ISO9001 il cui stato corrente coincide con *Audit*.

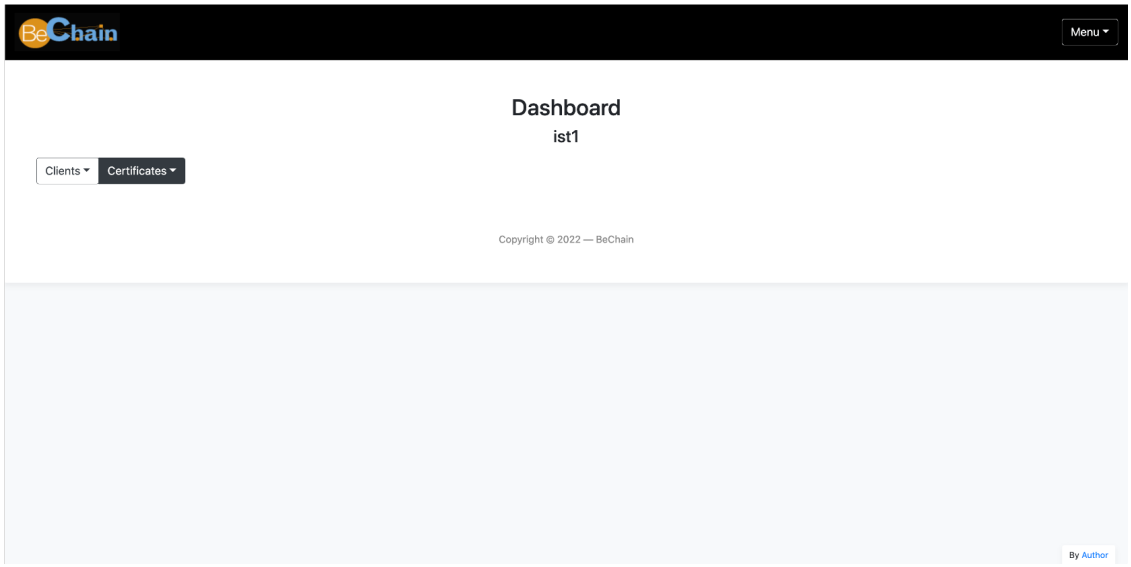

Figura 5.73. Dalla pagina personale clicchiamo *Certificates*.

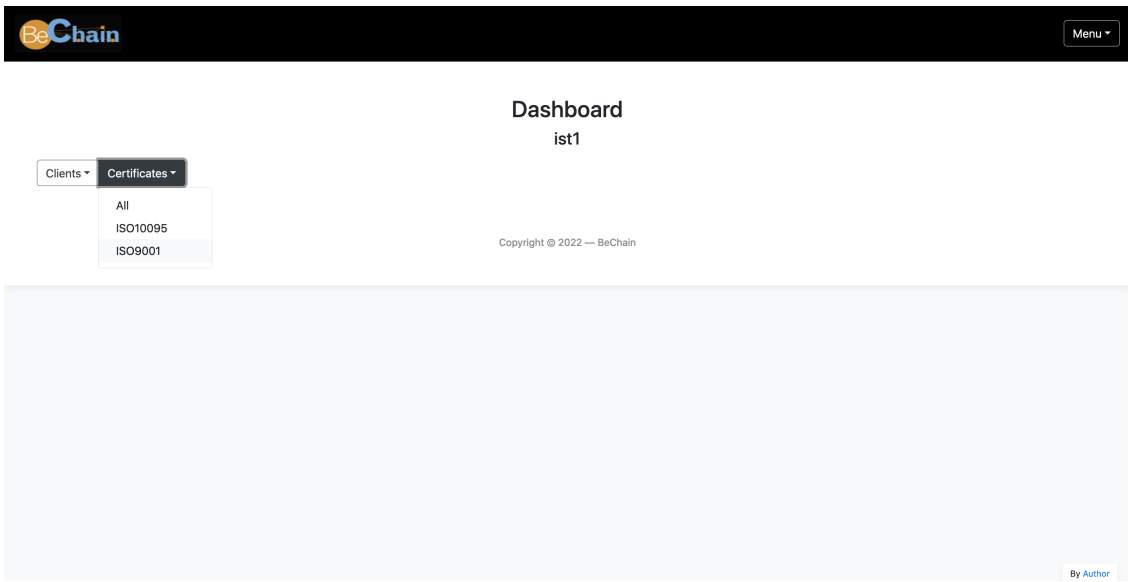

Figura 5.74. Clicchiamo su *ISO9001*.

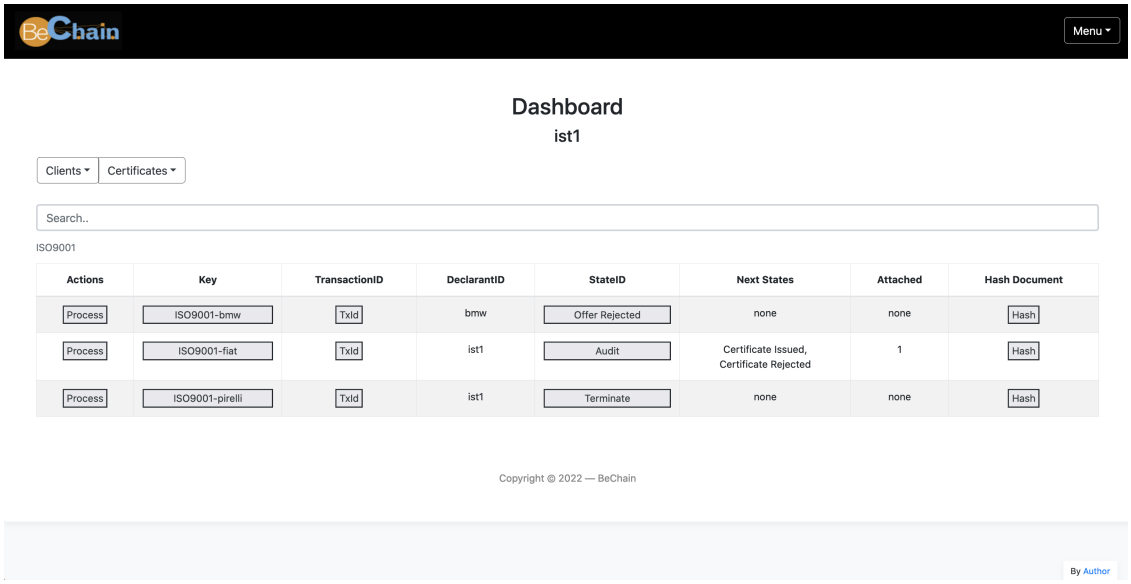

Figura 5.75. Visualizzazione di tutti i processi di certificazione relativi al contratto ISO9001.

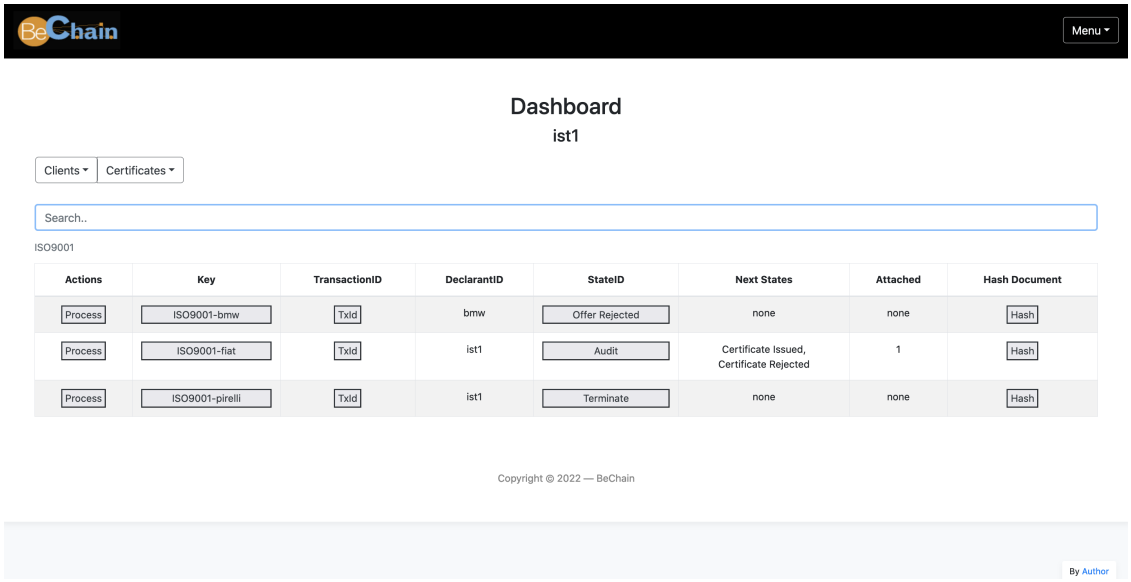

Figura 5.76. Selezioniamo la barra di ricerca in alto.

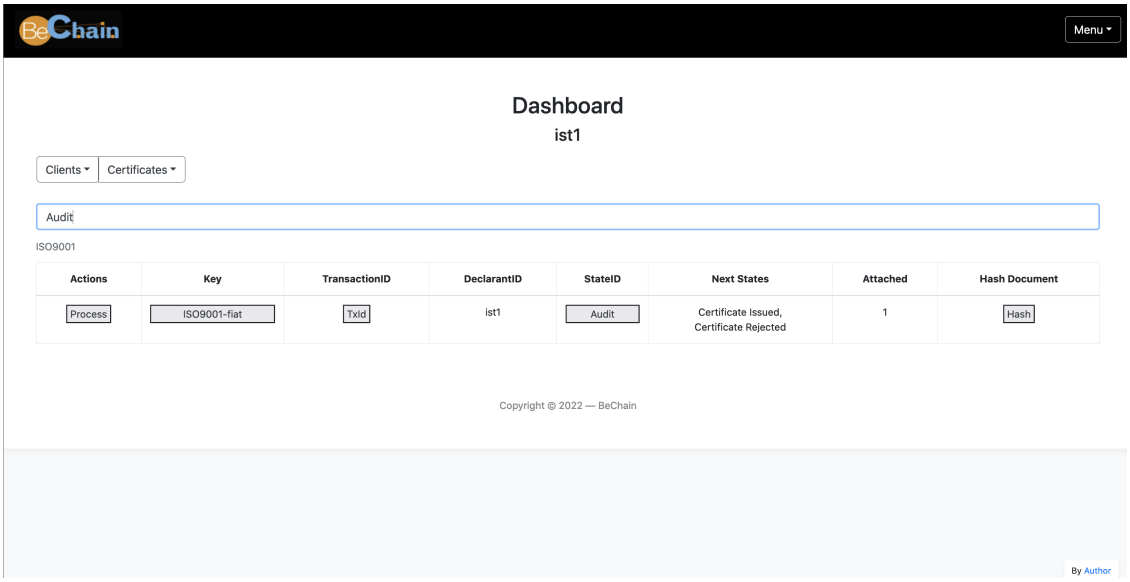

Figura 5.77. Digitiamo nella barra di ricerca la parola *Audit* e vediamo che nello stato mondiale è visualizzata un'unica transazione il cui *StateId* coincide con la chiave di ricerca.

Visualizzazione del processo di certificazione relativo a un determinato contratto agendo come cliente

In questo paragrafo sono indicati i vari passaggi per poter visualizzare il processo di certificazione relativo al contratto ISO9001 con l'*ist1*.

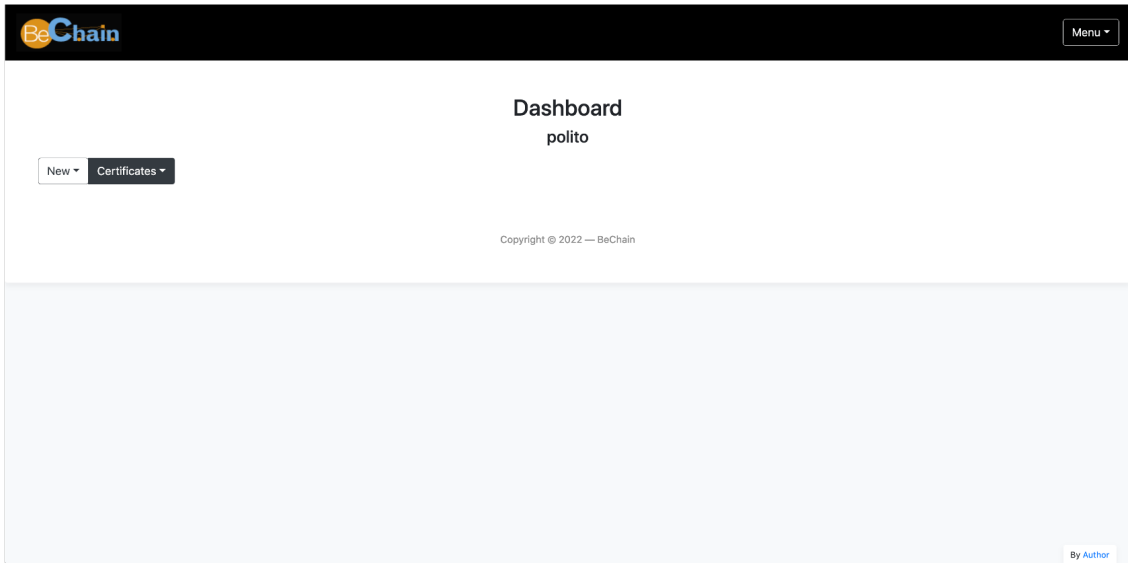

Figura 5.78. Dalla pagina personale clicchiamo *Certificates*.

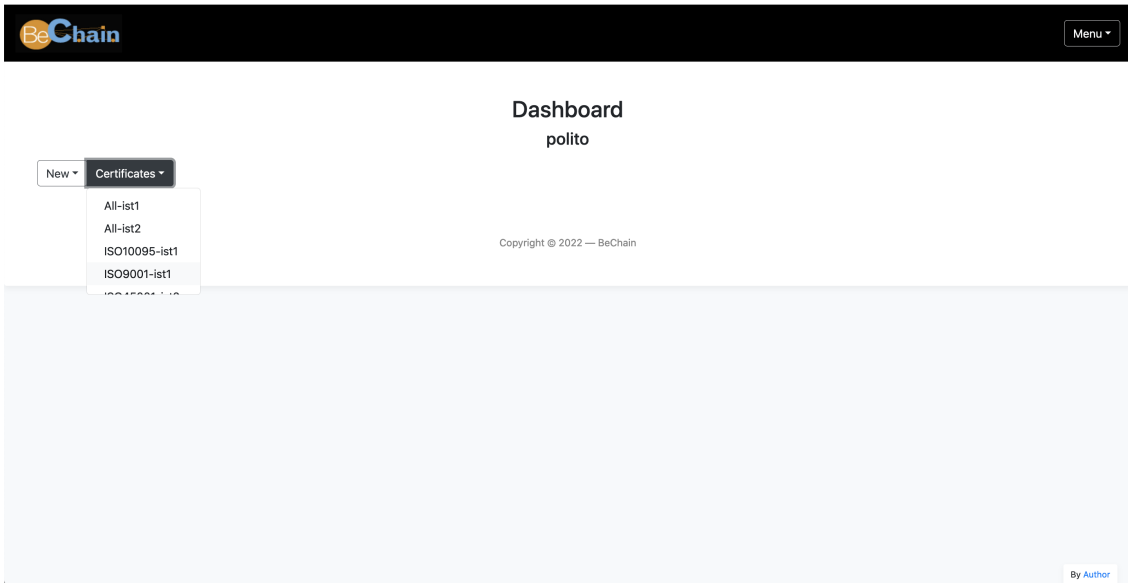

Figura 5.79. Clicchiamo su *ISO9001-ist1*.

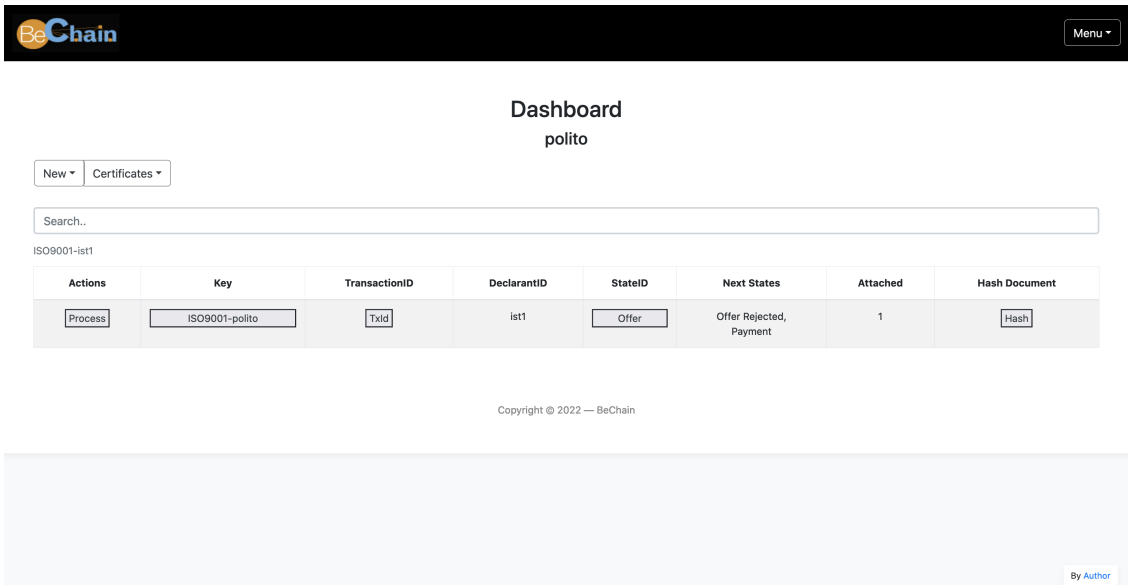

Figura 5.80. Visualizzazione del processo di certificazione relativo al contratto ISO9001 con l'*ist1*.

Visualizzazione di tutti i processi di certificazione con una determinata istituzione agendo come cliente

In questo paragrafo sono indicati i vari passaggi per poter visualizzare tutti i processi di certificazione con una determinata istituzione.

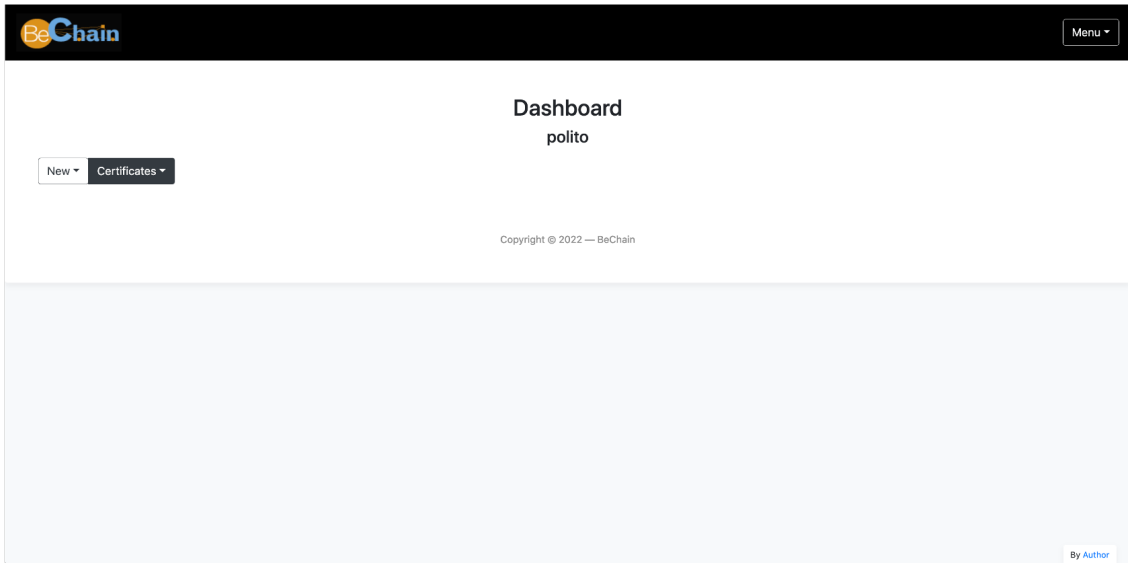

Figura 5.81. Dalla pagina personale clicchiamo su *Certificates*.

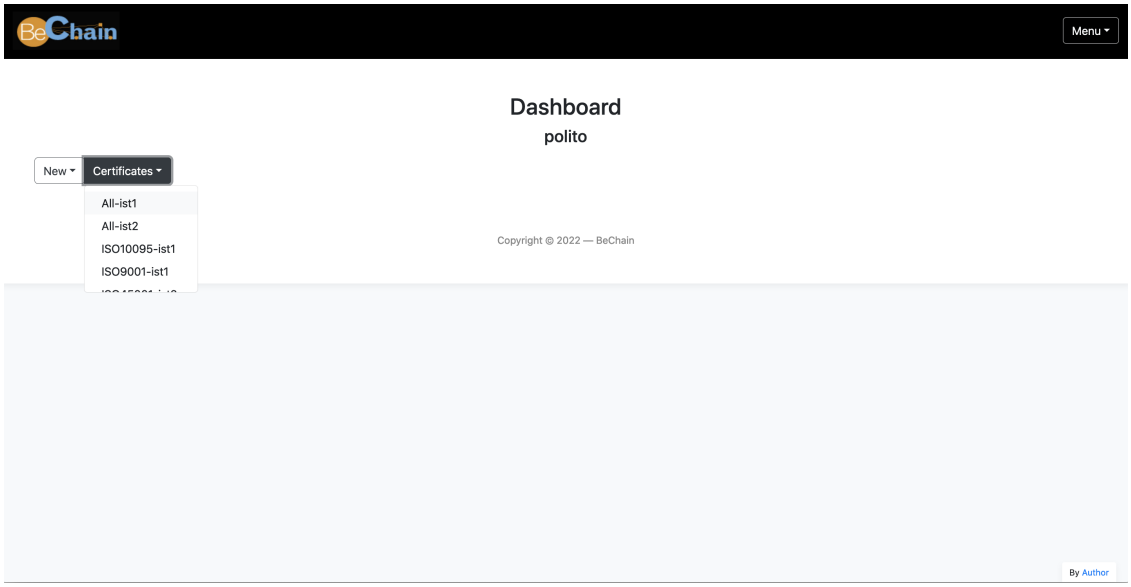

Figura 5.82. Clicchiamo su *All-ist1*.

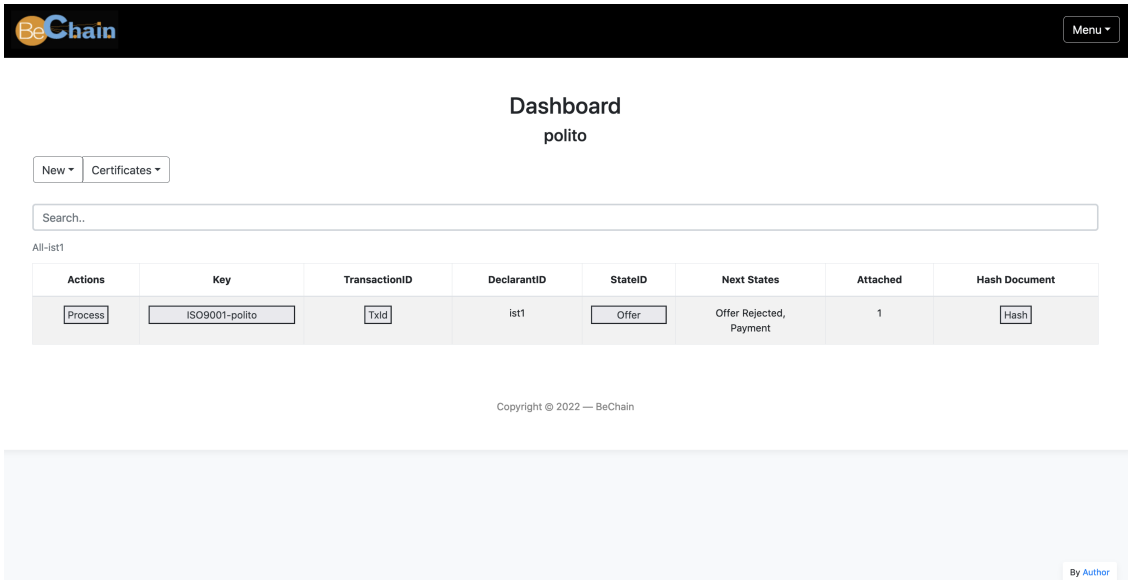

Figura 5.83. Visualizzazione di tutti i processi di certificazione tra *polito* e l'*ist1*.

Login

In questo paragrafo vediamo come un'istituzione effettua il login. I passaggi sono analoghi anche per il cliente.

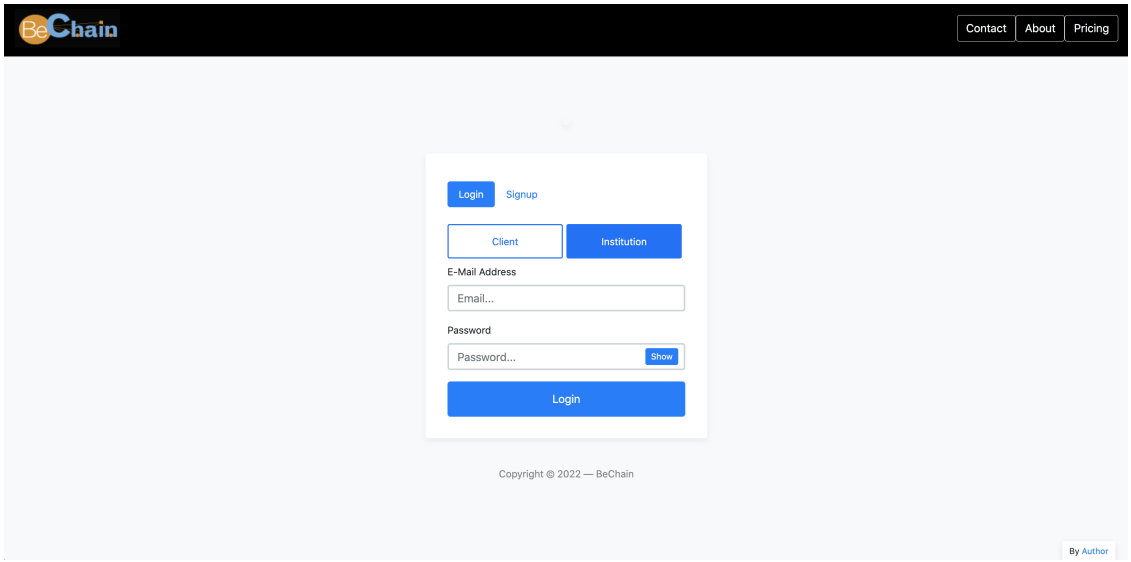

Figura 5.84. Form di login. Selezioniamo l'opzione *Institution*.

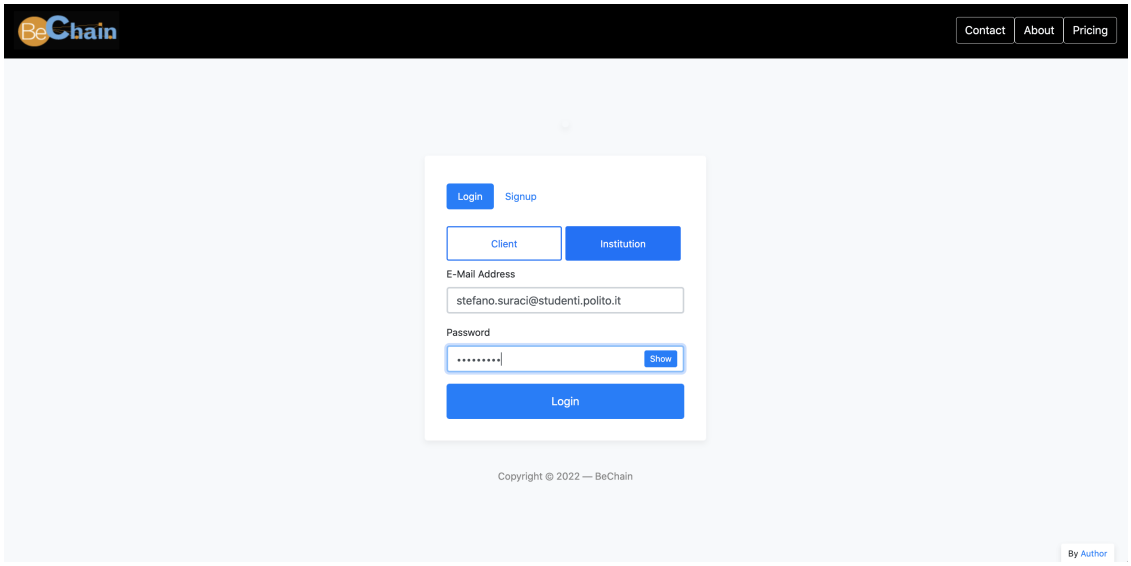

Figura 5.85. Digitiamo l'email e la password.

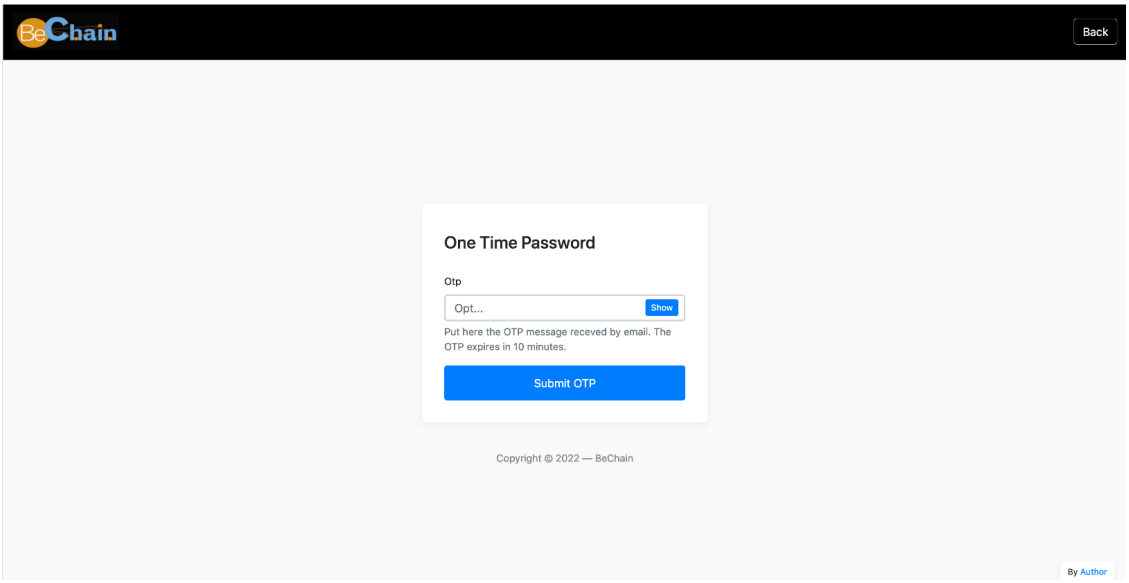

Figura 5.86. Form password monouso.

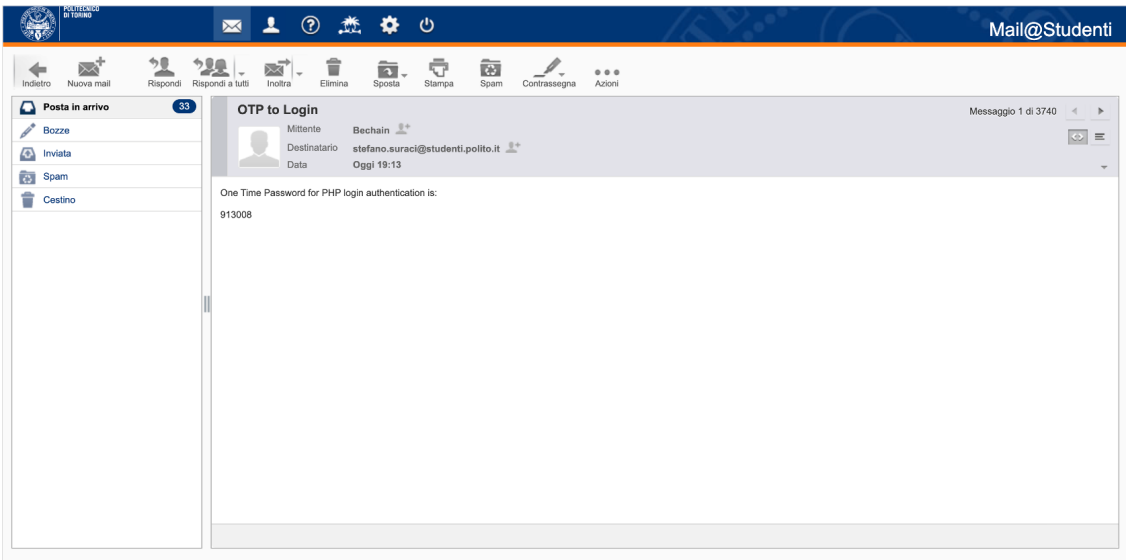

Figura 5.87. Visualizzazione della password monouso ricevuta nella propria casella di posta.

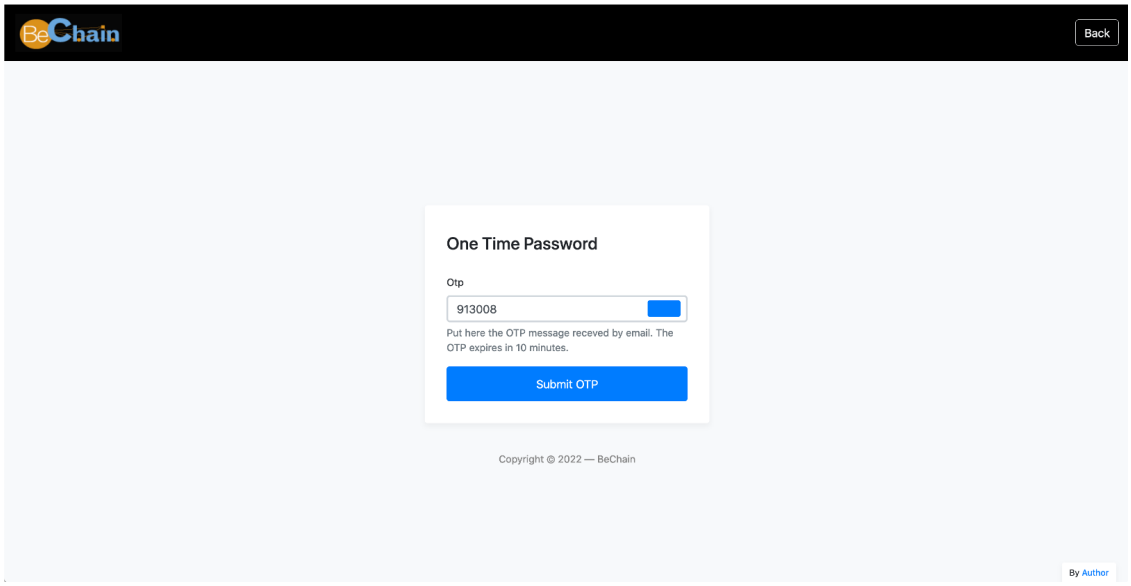

Figura 5.88. Inseriamo la password monouso all'interno del form *One Time Password*.

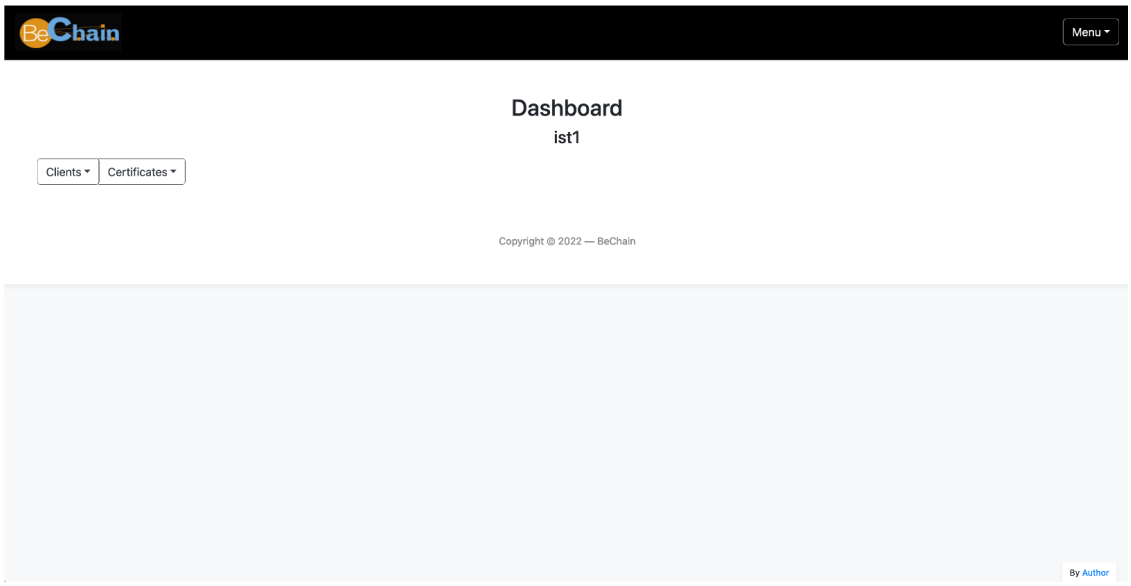

Figura 5.89. Se il login è avvenuto con successo si viene automaticamente indirizzati alla propria pagina personale.

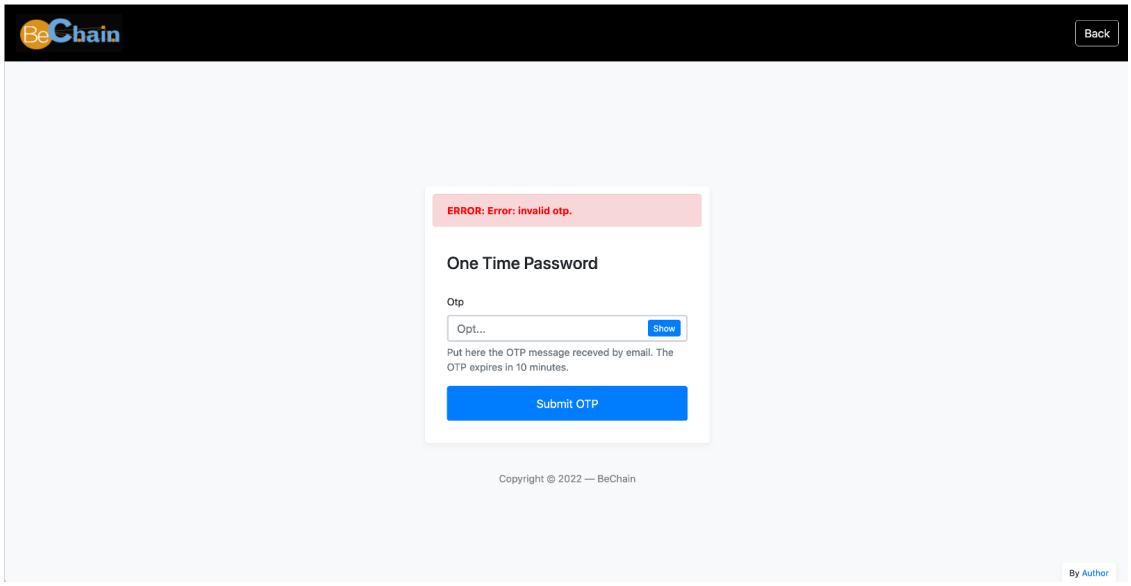

Figura 5.90. Se al contrario il login non è avvenuto con successo viene mostrato un messaggio di errore.

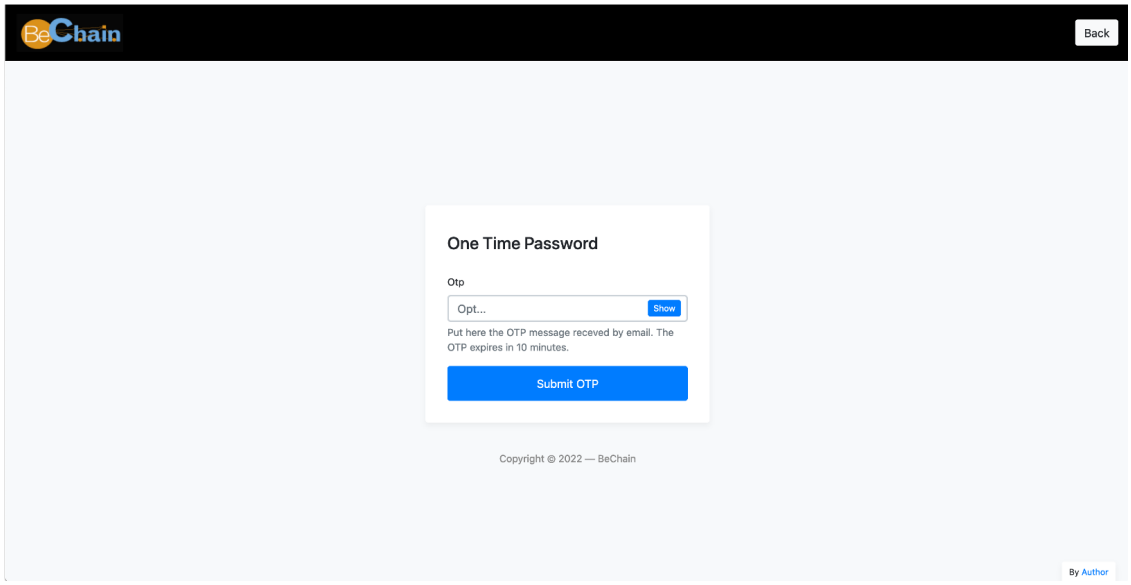

Figura 5.91. Cliccando sul pulsante *Back* è possibile tornare indietro alla pagina di login.

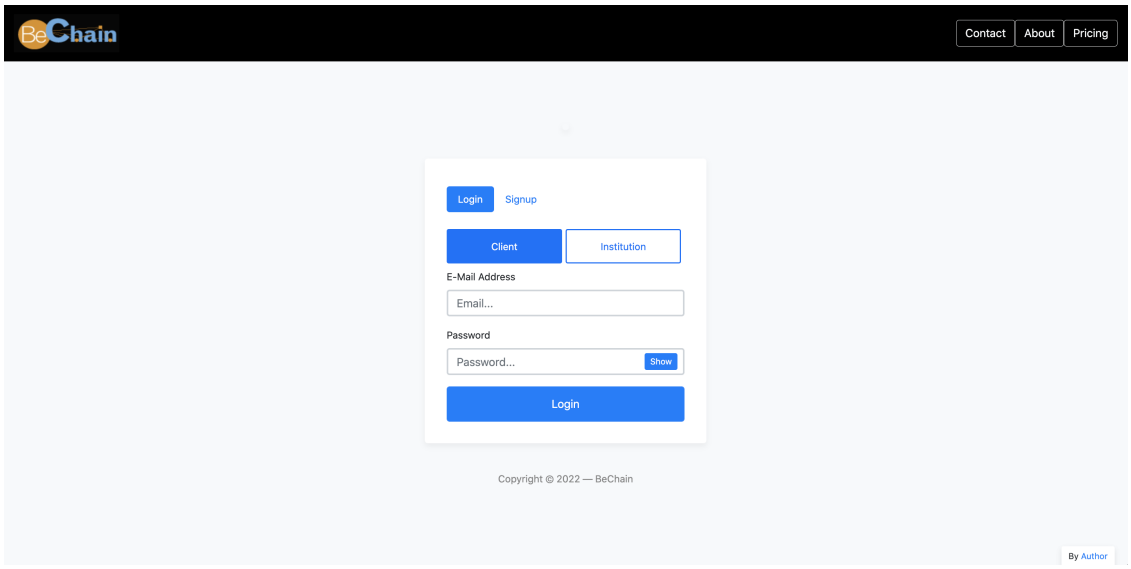

Figura 5.92. Form di login.

Logout

In questo paragrafo sono indicati i vari passaggi per poter effettuare il logout come istituzione. Il procedimento è analogo per un cliente.

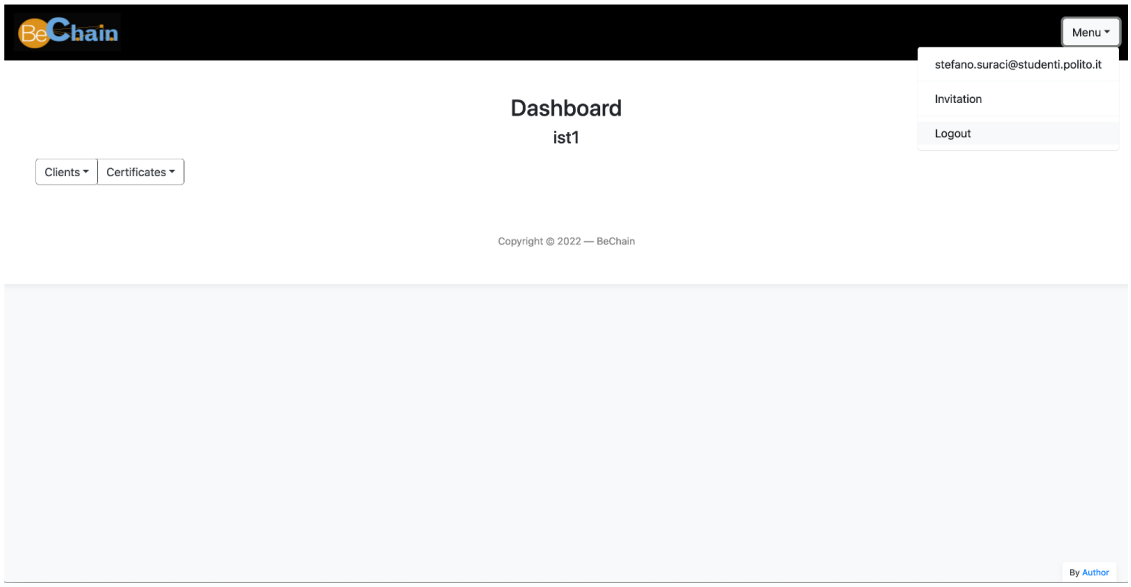

Figura 5.93. Clicchiamo su *Logout*.

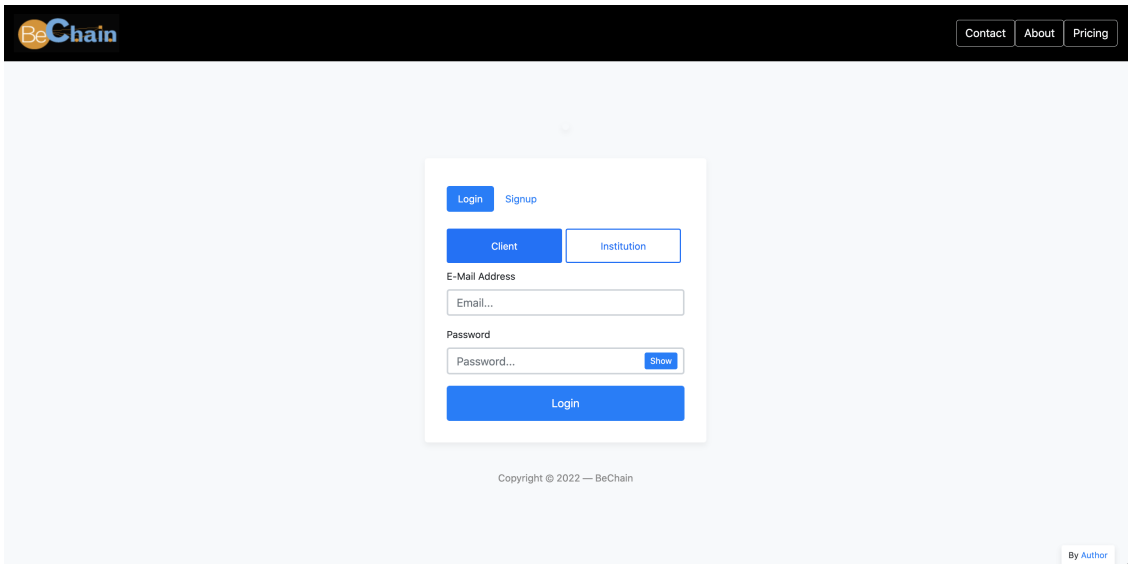

Figura 5.94. Automaticamente si viene indirizzati al form di login.

5.1.2 Gestione degli errori

Errore: il tempo per annullare l'ultima transazione è scaduto

In questo paragrafo viene mostrato come il sistema gestisce il tentativo di annullare una transazione dopo due muniti dalla sua esecuzione. Il sistema cataloga questo tipo transazione come una violazione degli accordi stabiliti.

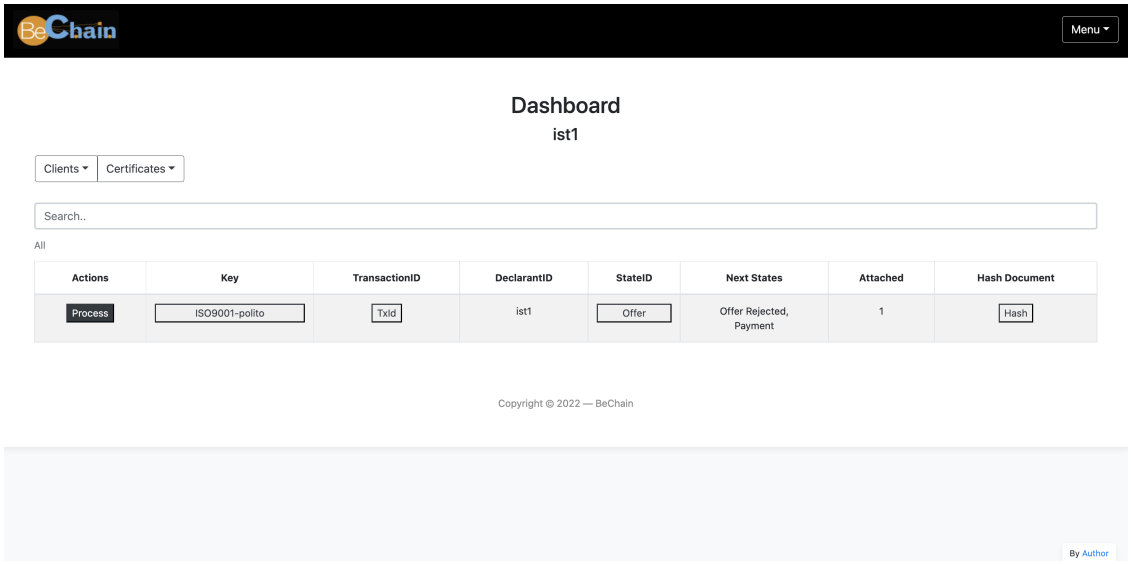

Figura 5.95. Clicchiamo su *Process*.

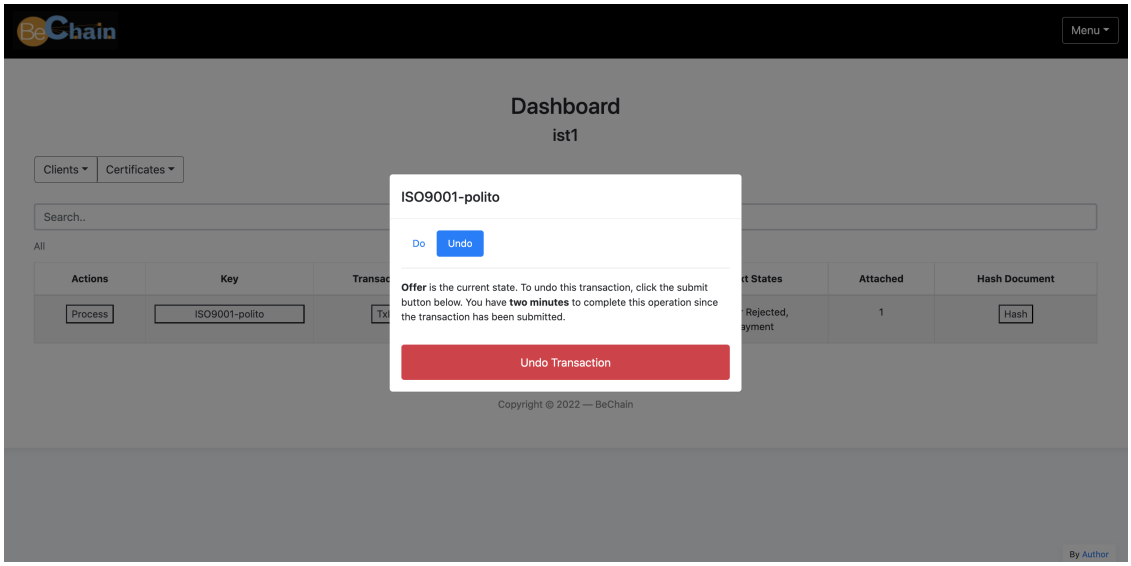

Figura 5.96. Clicchiamo su *Undo* e successivamente su *Undo Transaction*.

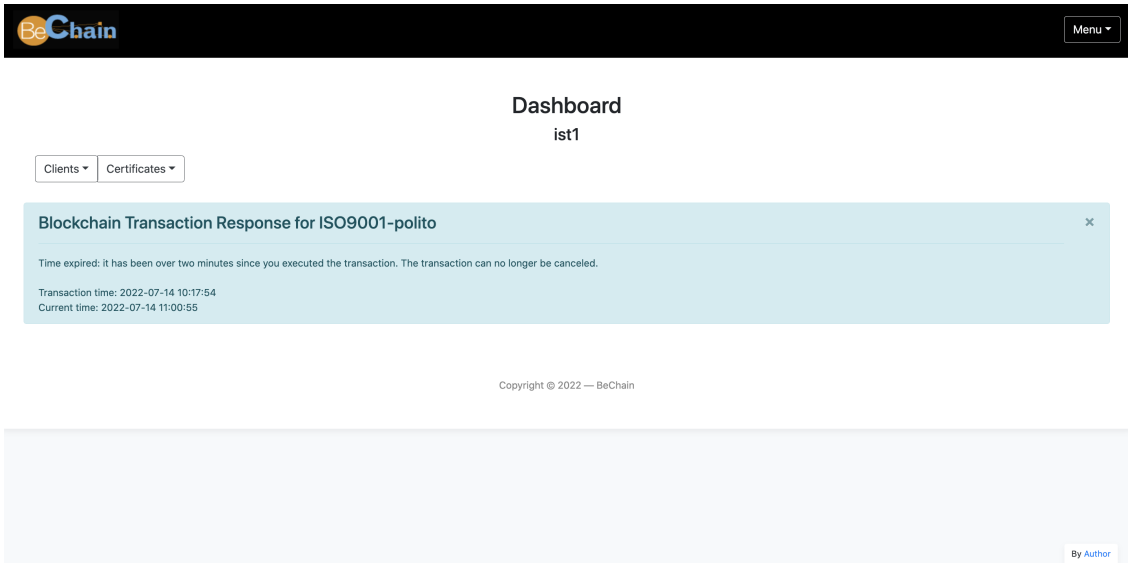

Figura 5.97. La transazione non è stata eseguita con successo. Il sistema mostra un messaggio di errore con la marca temporale dell'ultima transazione e dell'orario corrente.

Errore: impossibile annullare una transazione di cui non si è responsabili

In questo paragrafo viene mostrato come il sistema gestisce il tentativo di annullare una transazione che effettuata dalla controparte.

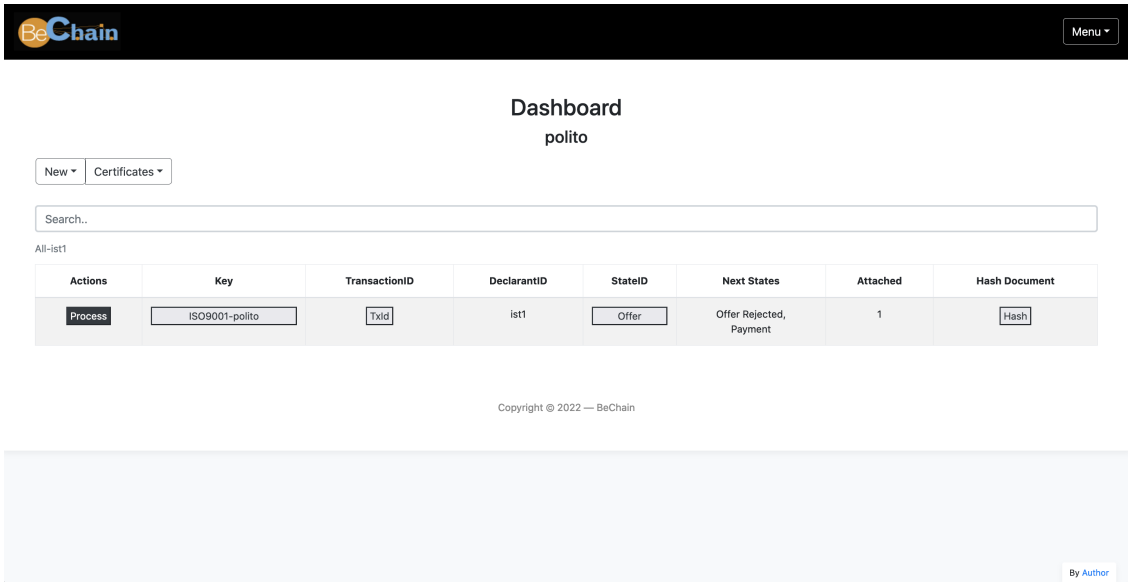

Figura 5.98. Clicchiamo su *Process*.

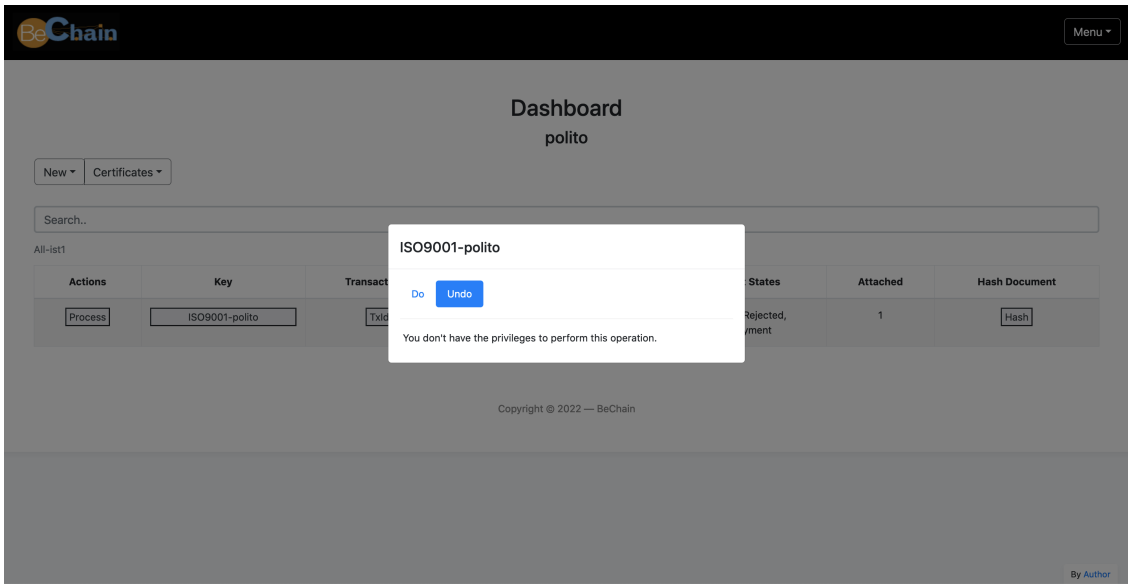

Figura 5.99. Il messaggio mostra l'impossibilità di annullamento.

Errore: documento richiesto

In questo paragrafo viene mostrato come il sistema gestisce il tentativo di eseguire una transazione senza allegare la documentazione richiesta.

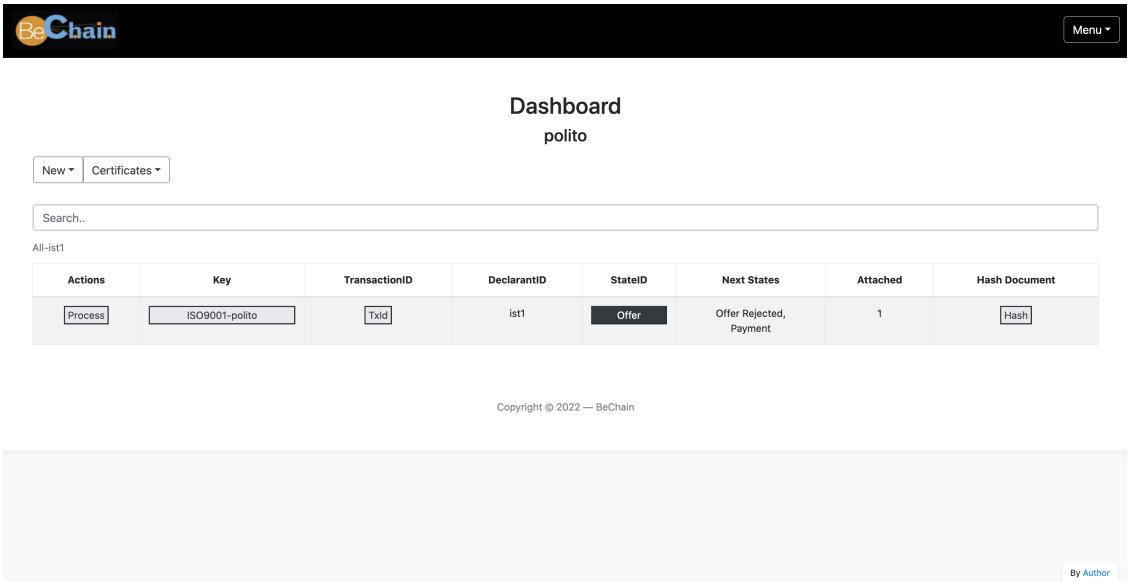

Figura 5.100. Clicchiamo sullo stato corrente.

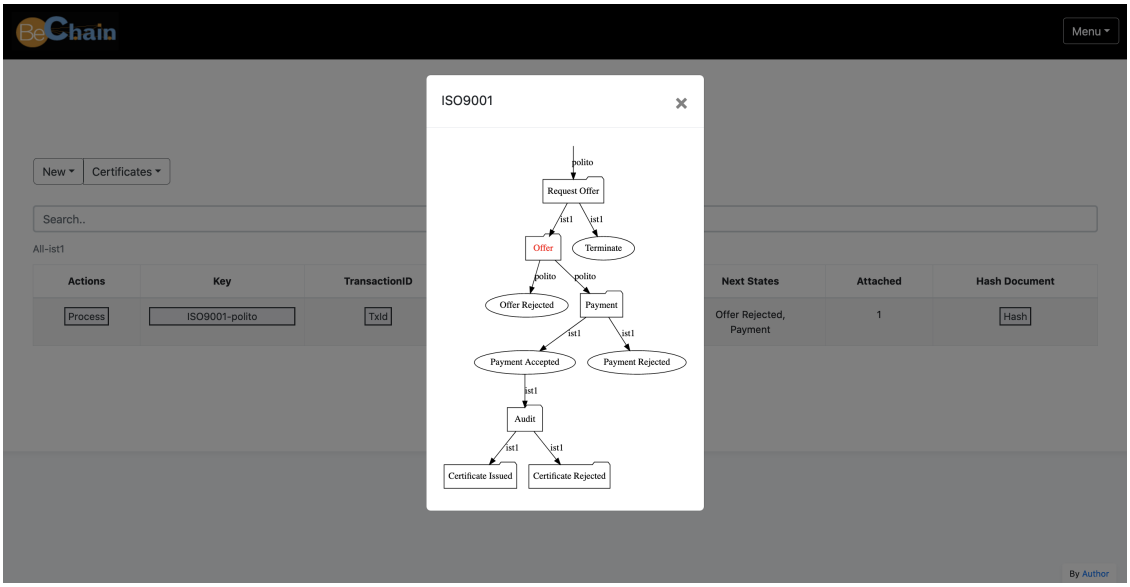

Figura 5.101. La rappresentazione grafica della macchina a stati mostra che la transazione relativa al pagamento richiede di fornire la documentazione.

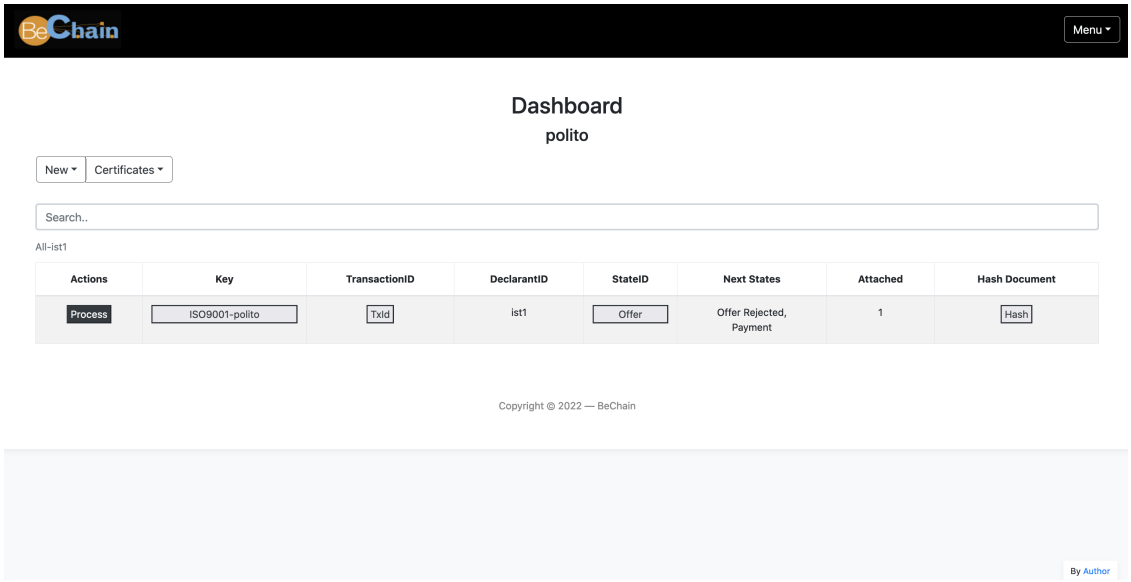

Figura 5.102. Clicchiamo su *Process*.

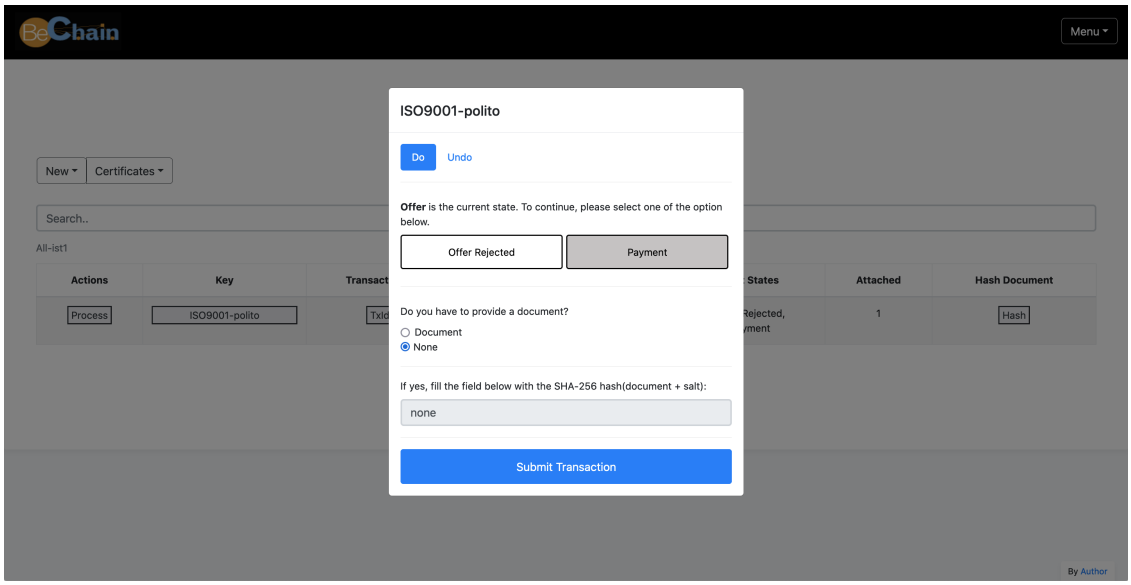

Figura 5.103. Selezioniamo *Payment* e successivamente *None*. In questo modo il campo relativo all'hash più sale diventa non editabile.

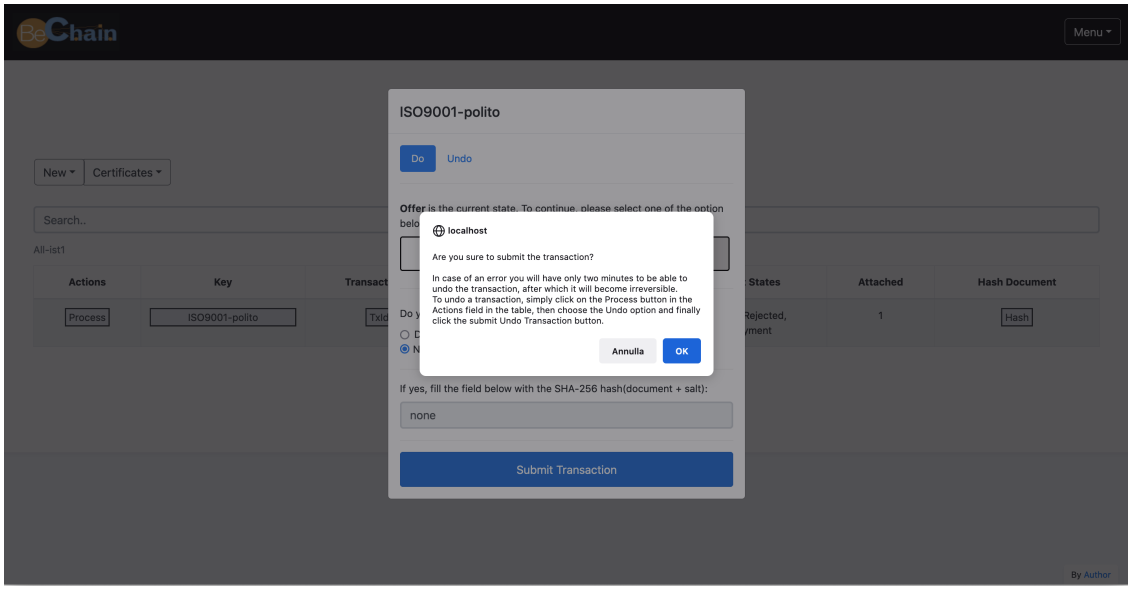

Figura 5.104. Confermiamo di voler eseguire la transazione.

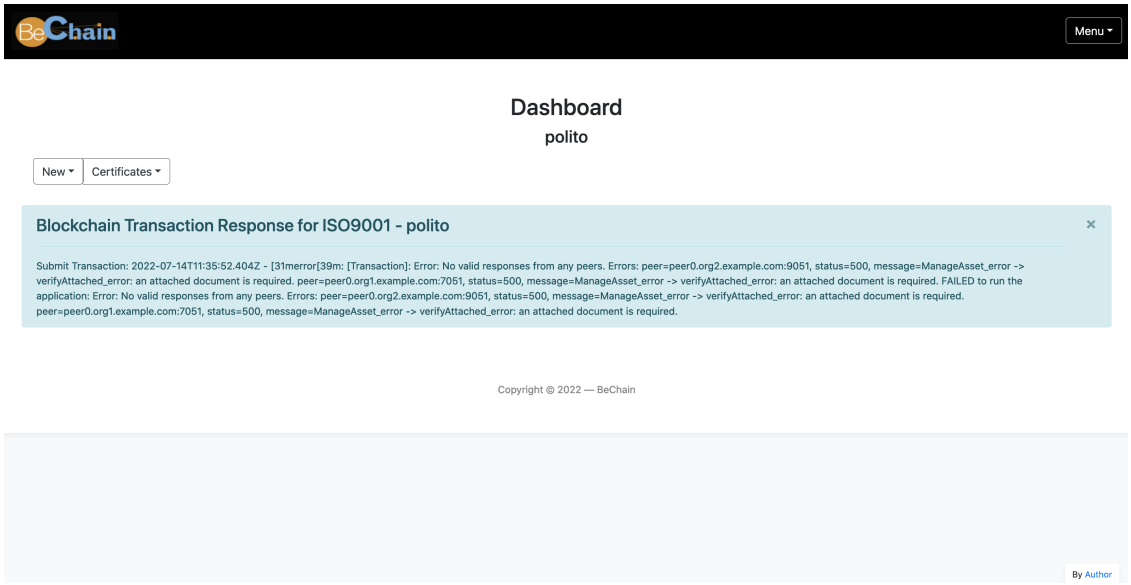

Figura 5.105. Transazione non eseguita. Il messaggio indica che è necessario fornire un documento.

Errore: documento non richiesto

In questo paragrafo viene mostrato un caso speculare al precedente. Un utente tenta di eseguire una transazione allegando un documento non richiesto.

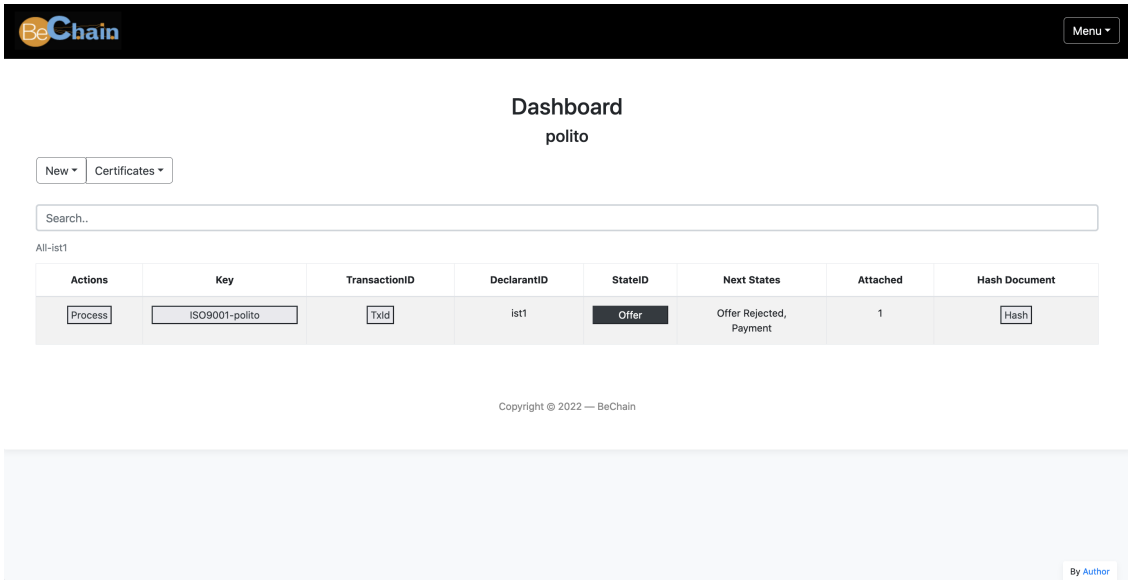

Figura 5.106. Clicchiamo sullo stato corrente.

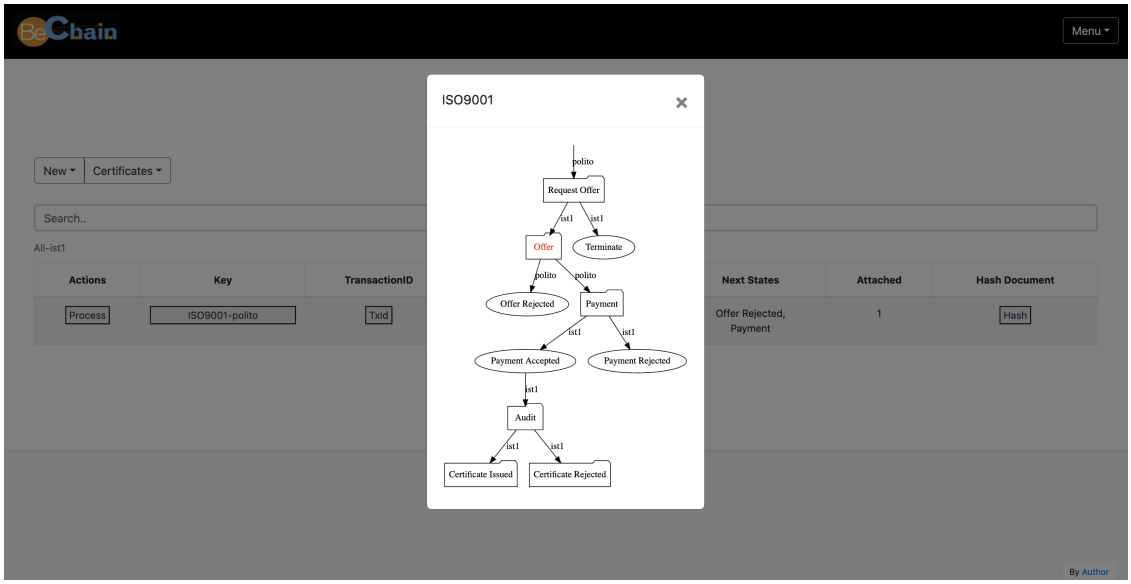

Figura 5.107. La rappresentazione grafica della macchina a stati mostra che la transazione relativa a *Offer Reject* non richiede documentazione.

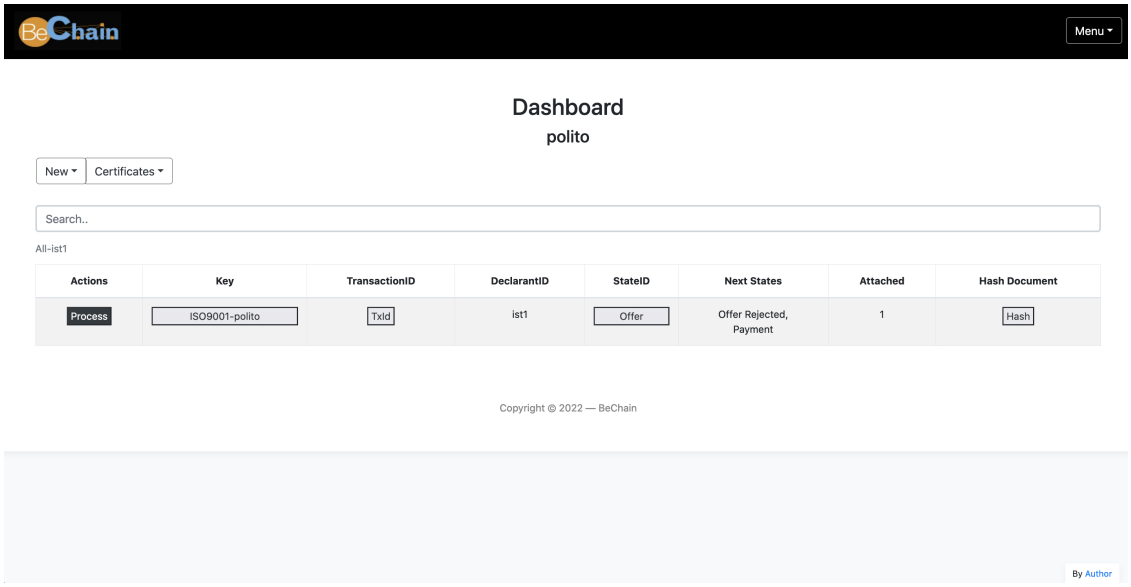

Figura 5.108. Clicchiamo su *Process*.

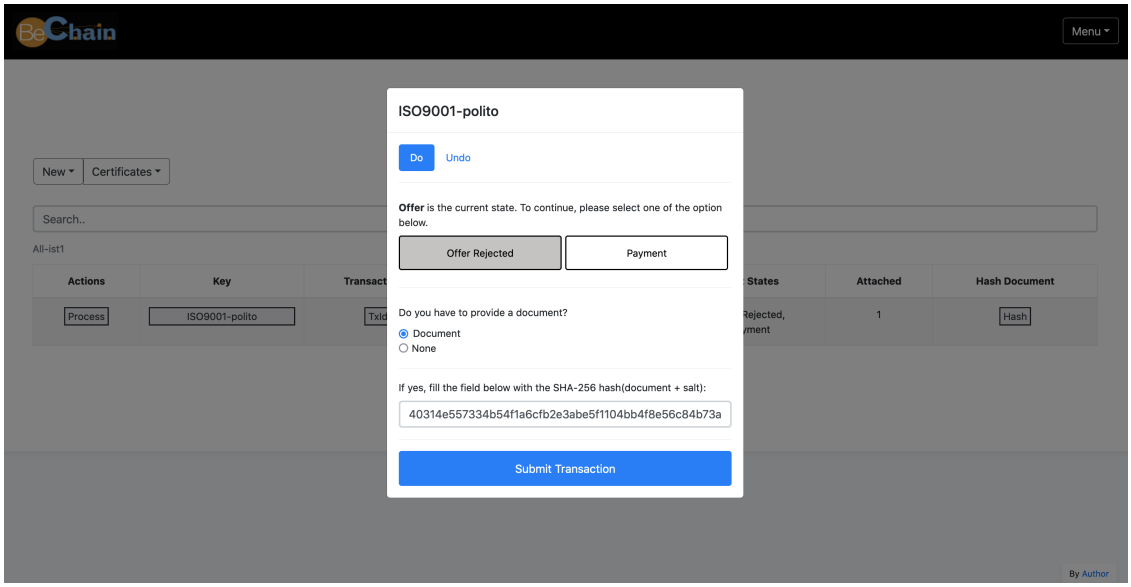

Figura 5.109. Selezioniamo *Offer rejected* e digitiamo l'hash più il sale del documento.

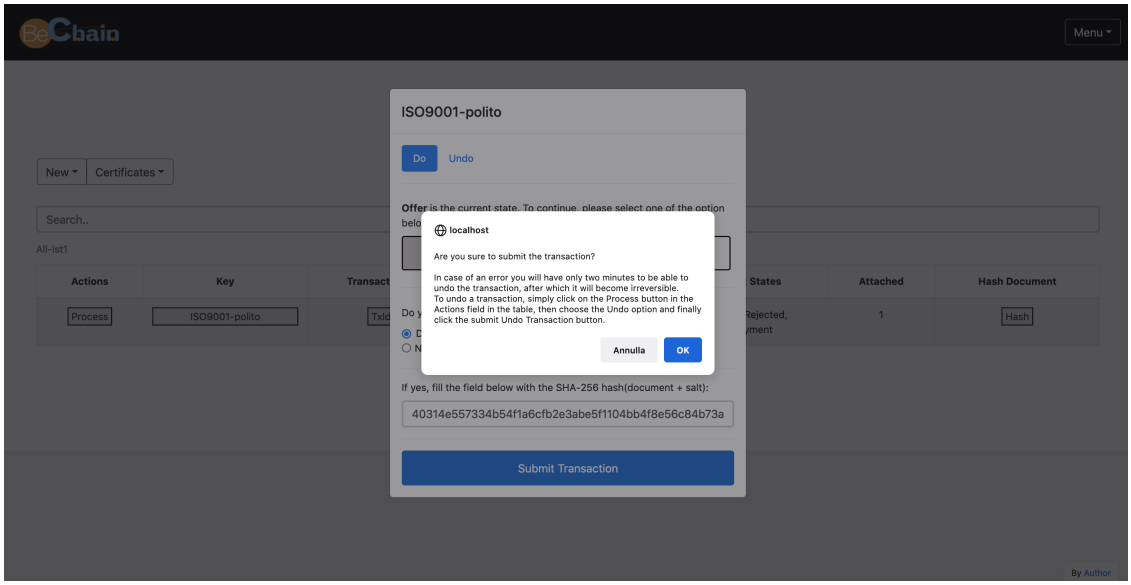

Figura 5.110. Confermiamo di voler eseguire la transazione.

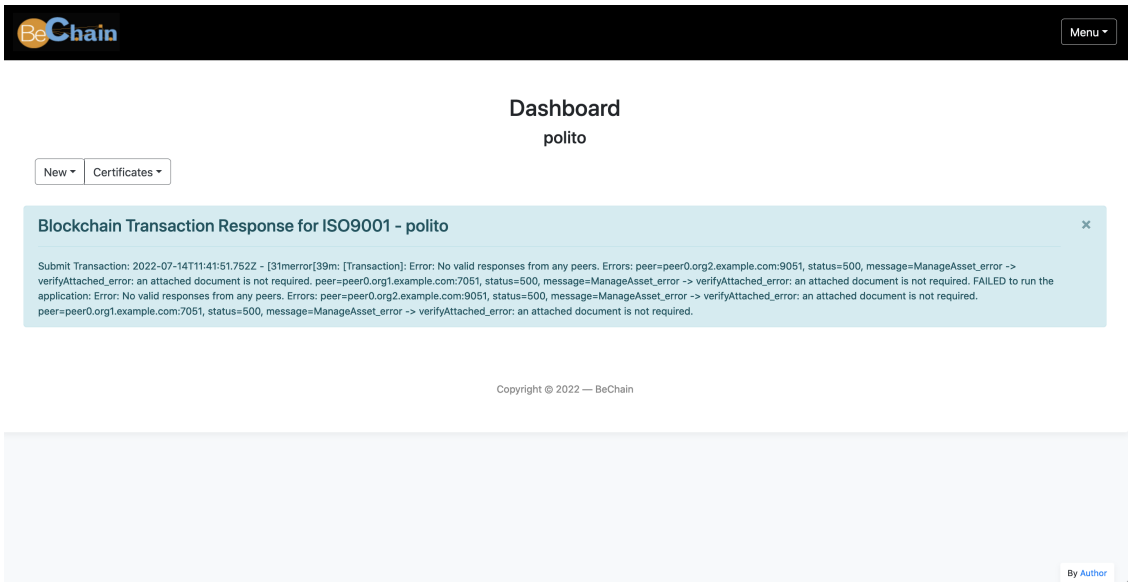

Figura 5.111. Transazione non eseguita. Il messaggio indica che non è necessario allegare un documento.

Errore: non è possibile eseguire una transazione se non si ha l'autorizzazione

In questo paragrafo viene mostrato il tentativo da parte di un'istituzione di svolgere un'azione assegnata a un cliente.

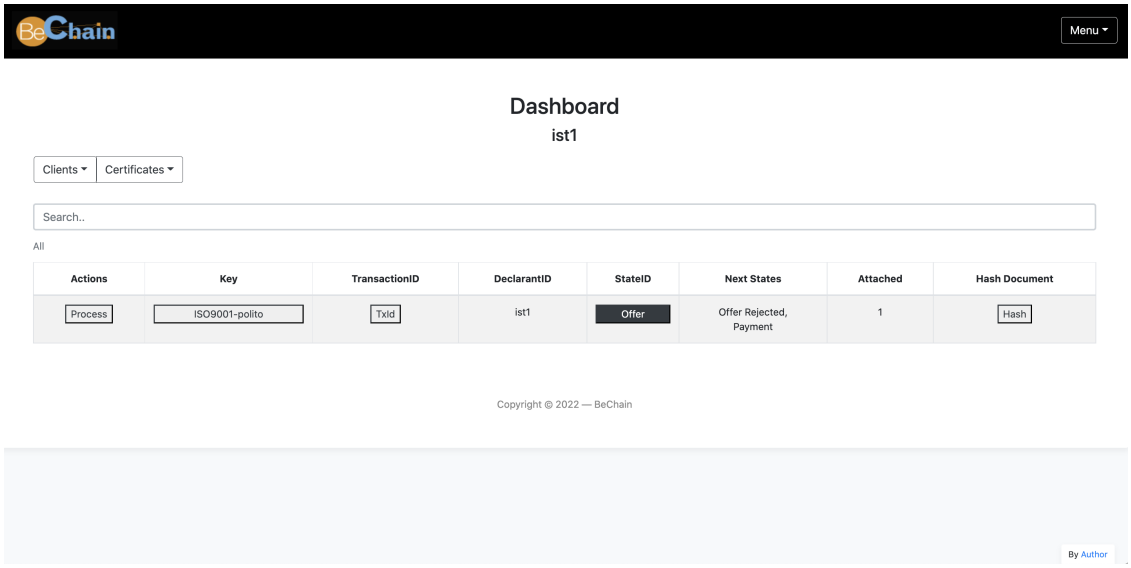

Figura 5.112. Clicchiamo sullo stato corrente.

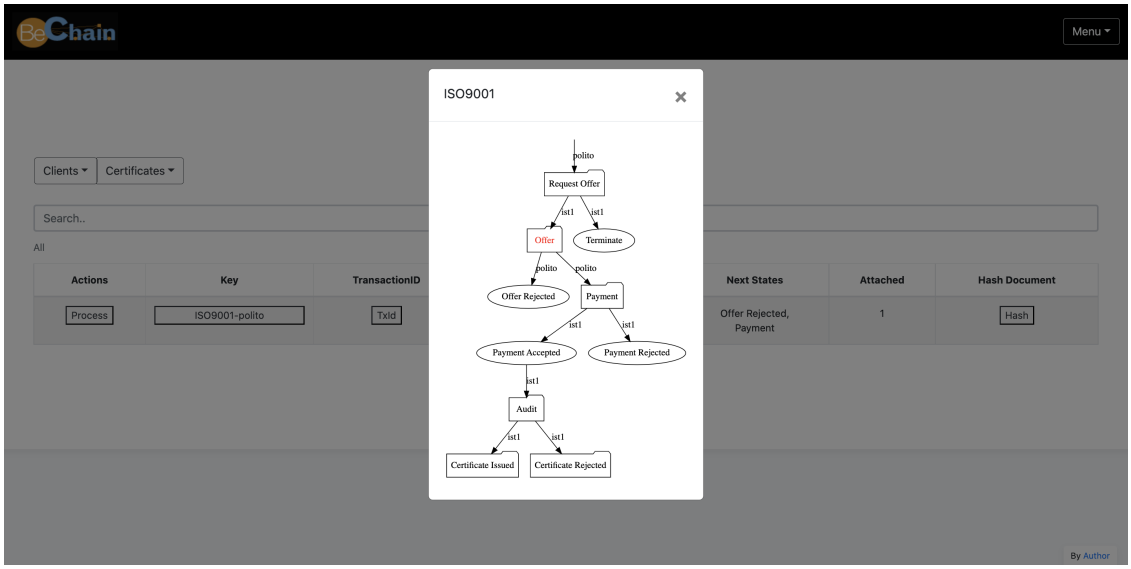

Figura 5.113. La rappresentazione grafica della macchina a stati mostra che la transazione di *Payment* deve essere eseguita da un cliente.

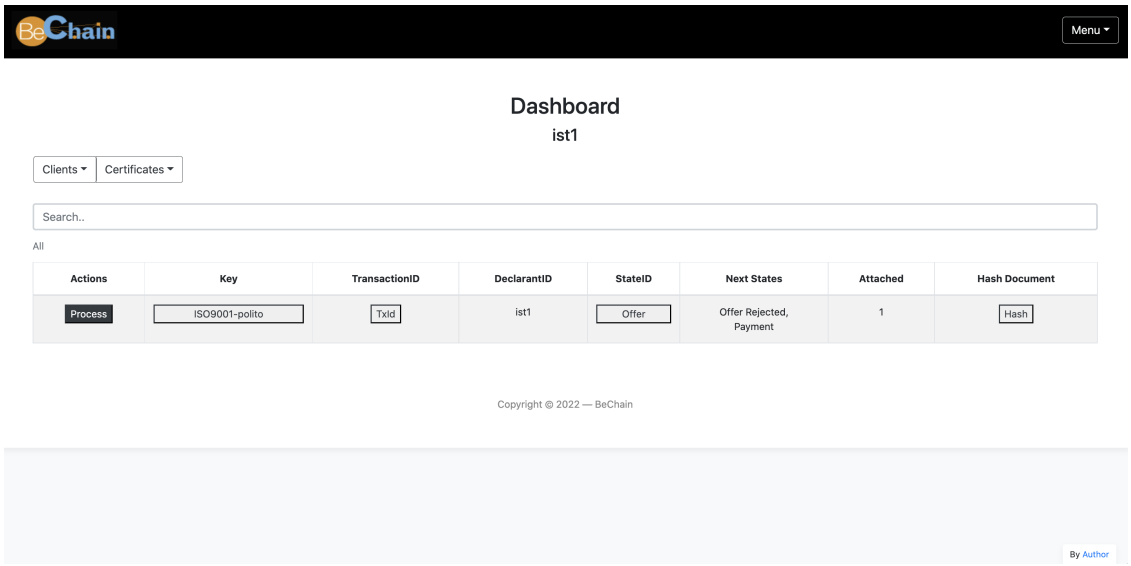

Figura 5.114. Clicchiamo su *Process*.

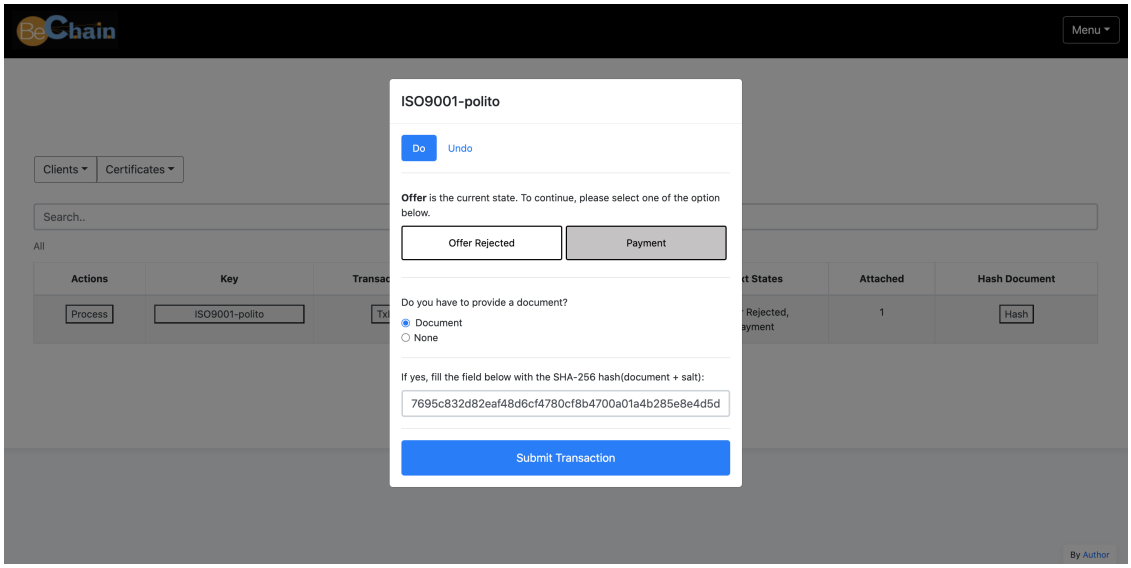

Figura 5.115. Selezioniamo *Payment* e digitiamo l'hash più il sale del documento.

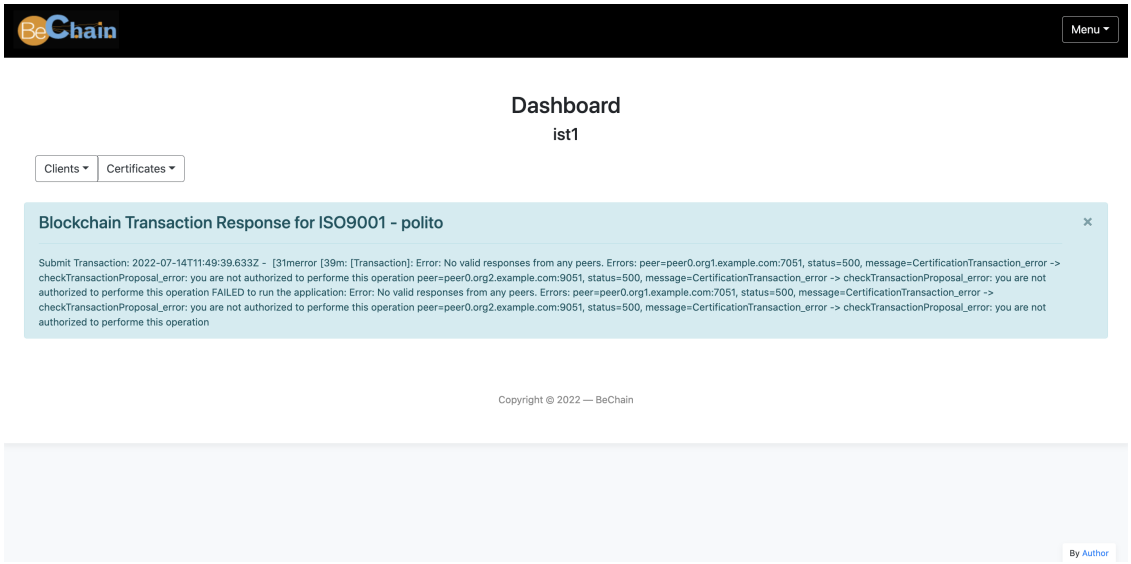

Figura 5.116. Transazione non eseguita. Il messaggio indica che non si ha l'autorizzazione.

Capitolo 6

Conclusioni

6.1 Conclusioni

Nel lavoro di tesi sono stati discussi diversi aspetti riguardante la tecnologia blockchain e il modello delle certificazioni. Confrontando i due modelli emerge in modo evidente il ruolo centrale occupato dalla fiducia in entrambi i sistemi di riferimento: la blockchain è la prima tecnologia in grado spostare il ruolo della fiducia dalle persone alle applicazioni; e questa transizione è mossa dalle necessità delle persone stesse e del loro bisogno di strumenti sempre più accurati e attendibili.

La scelta di utilizzare Hyperledger Fabric per la realizzazione dell'applicazione si è dimostrata efficace. Grazie a questa piattaforma sono stati garantiti i due principali requisiti di una blockchain aziendale: identificazione dei partecipanti e riservatezza delle informazioni. Inoltre, si è potuto godere delle proprietà di correttezza e incorruttibilità dei dati tipici di questa tecnologia.

Lo sviluppo degli applicativi di supporto hanno assegnato a Bechain una risorsa essenziale per poter modellare la logica aziendale e per velocizzare i tempi di sviluppo.

Infine, la realizzazione dell'applicazione web, ha fornito a enti di certificazione e clienti un'interfaccia grafica molto simile a quella di un'applicazione *blockchain explorer*, ma con l'aggiunta di funzionalità specifiche per il programma.

6.2 Sviluppi futuri

Dal punto di vista architetturale l'applicazione è stata realizzata considerando anche la possibilità di estensioni future. In questo paragrafo si cerca di rispondere a quella che potrebbe essere un'esigenza di Bechain in merito al servizio che offre. Si noti che questa architettura potrebbe essere implementata in qualsiasi momento nella rete attuale, proprio grazie alla modularità offerta dalla piattaforma Hyperledger Fabric.

Bechain è l'organizzazione che fornisce la propria infrastruttura blockchain a enti di certificazione e clienti. Per tale servizio Bechain potrebbe richiedere un compenso basato sull'utilizzo effettivo della rete. A tal proposito potrebbe richiedere alle istituzione un compenso basato sul numero di transazioni effettuate. Il motivo per cui Bechain dovrebbe coinvolgere solo le istituzioni è motivato dalle seguenti riflessioni: la prima è che Bechain

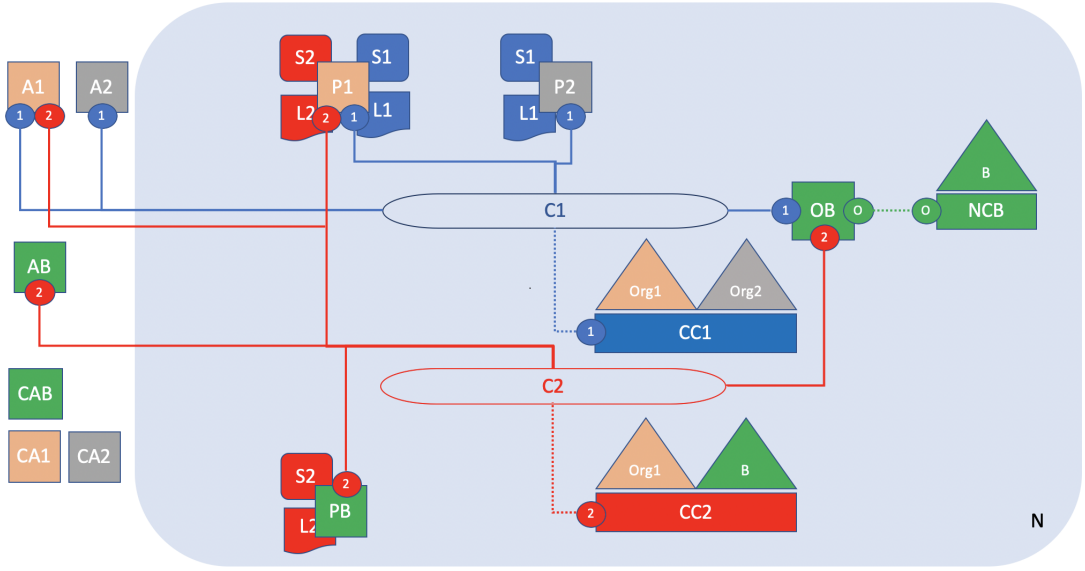

Figura 6.1. Rispetto all'architettura base, la rete contiene ora un nuovo consorzio composto dall'Org1 e B. Le due organizzazioni possono comunicare in modo privato tramite il canale C2 e agiscono secondo le regole del canale CC2. B ha un proprio *peer* PB e sia PB che P1 contengono una copia del *ledger* L2 e del *chaincode* S2. Le applicazioni *client* A1 e AB permettono di invocare gli *smart contracts* definiti nel *chaincode* S2.

attualmente stipula accordi solamente con le istituzioni, mentre le negoziazioni tra enti di certificazione e clienti sono gestite autonomamente dalle parti interessate. In secondo luogo, l'interesse principale dei clienti è ottenere il certificato e l'utilizzo della blockchain non dovrebbe modificare la loro esperienza utente.

Osserviamo ora l'architettura in figura. B crea un nuovo consorzio che include l'Org1 e B stesso. Successivamente viene creato il canale C2 per permettere una comunicazione privata tra l'Org1 e B. Questo canale ha una configurazione CC2 che è totalmente distinta dalla configurazione dell'altro canale CC1 e dalla configurazione di rete NCB. Sul canale C2 Bechain e i membri della prima organizzazione hanno uguali privilegi, mentre i membri della seconda organizzazione non hanno alcun diritto. A questo punto, affinché B possa utilizzare le funzionalità offerte dal canale C2, deve aggiungere il nodo *peer* PB che ospiterà il *chaincode* S2 e l'istanza del *ledger* L2. In questo caso L2 è un *ledger* completamente separato da L1 proprio perché il canale crea una comunicazione riservata ai soli membri costituenti. Infine, è necessario aggiungere l'applicazione *client* AB per poter invocare i contratti intelligenti definiti nel *chaincode* S2 e generare transazioni sul registro. Come mostrato in figura, l'applicazione client A1 deve poter utilizzare sia il canale C1 che il canale C2, pertanto il *peer* P2 ospiterà anche l'istanza del *ledger* L2 e il *chaincode* S2. L'Org1 è quindi un'organizzazione particolare perché è in grado di effettuare transazioni con entrambi i canali applicativi. A questo punto è importante sottolineare che il comportamento del *peer* dipende comunque dalle politiche del canale a cui in quel momento è sottoposto. In particolare è lecito aspettarsi che CC1 e CC2 siano

diverse in quanto rispecchiano le regole concordate tra diversi gruppi di organizzazioni.

Ciò che rende possibile questa architettura è la possibilità offerta da Hyperledger Fabric di poter effettuare interazioni di tipo *chaincode to chaincode*. Nel caso delle interazioni di *query*, il target del *chaincode* del chiamante può essere anche su un altro canale, mentre nel caso di transazioni di aggiornamento è necessario che il chiamante invochi il *chaincode* presente sullo stesso canale.

Bibliografia

[1] Ankur Sharma Felix Martin Schuhknecht Divya Agrawal Jens Dittrich. How to databasify a blockchain: the case of hyperledger fabric. https://arxiv.org/pdf/1810.13177v1.pdf, 2018. [Online; in data 31-ottobre-2018].

[2] Raffaele Bianchi Gianluca Chiap, Jacopo Ranalli. *Blockchain, Tecnologia e applicazioni per il business*. Ulrico Hoepli Milano, 2019. [Online; in data 15-marzo-2019].

[3] ISO. Iso strategy 2030. https://www.iso.org/files/live/sites/isoorg/files/store/en/PUB100364.pdf, 2021. [Online; 2021].

[4] Christian Gorenflo Stephen Lee Lukasz Golab S. Keshav. Fastfabric: Scaling hyperledger fabric to 20,000 transactions per second. https://arxiv.org/pdf/1901.00910.pdf, 2019. [Online; in data 3-gennaio-2019].

[5] Satoshi Nakamoto. Bitcoin: A peer-to-peer electronic cash system. https://bitcoin.org/bitcoin.pdf, 2008. [Online; in data 31-ottobre-2008].

[6] Pingcheng Ruan Tien Tuan Anh Dinh Dumitrel Loghin Meihui Zhang Gang Chen Qian Lin Beng Chin Ooi. Blockchain vs distributed database: Dichotomy and fusion. https://arxiv.org/pdf/1910.01310.pdf, 2021. [Online; in data 19-gennaio-2021].

[7] Filippo Trifiletti Emanuele Riva. Tecnologia blockchain in ambito di accreditamento nazionale. https://www.youtube.com/watch?v=wEzQnqBw6Ho&t, 2021. [Online; in data 13-luglio-2021].

[8] Accredia chi siamo. https://www.accredia.it/chi-siamo. [collegamento; Maggio 2022].

[9] Graphviz. https://graphviz.org. [collegamento; Maggio 2022].

[10] Hyperledger fabric key concepts. https://hyperledger-fabric.readthedocs.io/en/release-2.2/key_concepts.html. [collegamento; Maggio 2022].

[11] Hyperledger fabric whatis. https://hyperledger-fabric.readthedocs.io/en/release-2.2/whatis.html. [collegamento; Maggio 2022].

[12] Wikipedia. Problema dei generali bizantini — wikipedia, l'enciclopedia libera. http://it.wikipedia.org/w/index.php?title=Problema_dei_generali_bizantini&oldid=106176642, 2019. [Online; in data 28-giugno-2022].