POR-FESR EMILIA ROMAGNA 2014-2020

Asse 1 - Ricerca e innovazione

Azione 1.2.2 - Supporto alla realizzazione di progetti complessi di attività di ricerca e sviluppo su poche aree tematiche di rilievo e all'applicazione di soluzioni tecnologiche funzionali alla realizzazione della strategia di S3

Bando 2018

Progetti di ricerca industriale strategica rivolti agli ambiti prioritari della Strategia di Specializzazione Intelligente

Sistemi interoperabili ed efficienti per la gestione sicura di filiere industriali

Deliverable D3.1:
Progetto di alto livello di SmartChain per filiere

Data di consegna prevista:	31 Gennaio 2021
Autori:	CRIS, CIRI ICT, CROSSTEC, MECHLAV, TTLAB
Versione:	1

Indice

1. Introduzione 3
2. Criteri di progettazione 4
3. Progettazione del Sistema di tracciatura della filiera 5
   3.1. Modalità di protezione dei dati 8
   3.2. Funzionalità applicazione Web 10
   3.3. Funzionalità smart contract 12
   3.4. Gestione delle autorizzazioni e delle chiavi 14
   3.5. Flussi delle operazioni 15
4. Progettazione sistema Carpigiani 23
   4.1. Domini applicativi e componenti 25
   4.2. Funzionalità smart contract 27
   4.3. Funzionalità applicazione Web 29

1. Introduzione

In questo documento si descrive la progettazione di alto livello dei sistemi per il progetto SmartChain, nell'ambito della Fase 3 "Progettazione e realizzazione delle piattaforme". Questo documento utilizza come input i documenti che descrivono i risultati delle precedenti Fasi 1 e 2, e in particolare è direttamente legato al documento D2.2 "Specifiche del sistema SmartChain", da cui attinge le principali specifiche funzionali e non funzionali di progetto, compresi gli aspetti di modellazione degli scenari e dei casi d'uso affrontati, delle informazioni da gestire, degli attori che utilizzeranno il sistema, e di alcune preliminari valutazioni di sicurezza e di performance che si attendono dal sistema finale. Questo documento ha l'obiettivo di descrivere i risultati dell'attività di progettazione di alto livello dei sistemi descrivendo tutti gli aspetti che riguardano le principali scelte architetturali e alcune scelte tecnologiche. Non ha l'obiettivo di descrivere tutti gli aspetti implementativi, di configurazione o di sperimentazione che caratterizzeranno i sistemi finali, che verranno invece descritti nel documento D3.2 "Progetto di dettaglio di SmartChain per filiere".

Il documento è strutturato nelle seguenti modalità: nella Sezione 2 si presentano i criteri di progettazione impiegati nell'attività; nella Sezione 3 si descrivono i risultati dell'attività di progettazione per il caso d'uso Bianco Accessori; nella Sezione 4 si descrivono i risultati dell'attività di progettazione per il caso d'uso Carpigiani.

2. Criteri di progettazione

La progettazione è stata guidata da diversi criteri di progettazione che sono stati condivisi da entrambi i sistemi specializzati per i casi d'uso previsti dal progetto.

- Quando possibile le scelte progettuali considerano di utilizzare software open source (*free software*), soprattutto nel caso dei componenti che gestiscono aspetti critici nell'ambito della sicurezza del sistema
- Il sistema è pensato per essere modulare, potenzialmente modificabile in alcune sue parti o estendibile con ulteriori componenti
- Il sistema è pensato per essere scalabile e considerare costi di gestione e mantenimento
- Nella scelta delle tecnologie si è deciso di privilegiare quelle che garantiscono maggiore affidabilità e un orizzonte temporale di utilizzo molto ampio in particolare per quanto riguarda le piattaforme blockchain (anche alla luce del costo tecnologico di realizzazione dell'infrastruttura). Per fare ciò ci si è orientati in particolare sulle soluzioni che vedono il coinvolgimento di partner solidi nel loro sviluppo e che risultano già essere utilizzate con successo in contesti reali,
- Per favorire il coinvolgimento di quei partner che non dispongono di un reparto IT particolarmente evoluto, si è deciso di puntare a una integrazione facilitata dei sistemi blockchain all'interno dell'architettura sfruttando strumenti quali la realizzazione di macchine virtuali adatte all'esecuzione sui più comuni servizi cloud presenti nel mercato.
- Per l'accesso delle informazioni all'utente si è deciso di prediligere l'utilizzo di interfacce grafiche di tipo Web in quanto risultano facilmente utilizzabili anche da personale non tecnico utilizzando dispositivi di uso comune per la loro fruizione.

3. Progettazione del Sistema di tracciatura della filiera

Si descrive l'architettura di alto livello del sistema di tracciatura della filiera, che ha come riferimento il caso d'uso dell'azienda Bianco Accessori, tramite la Figura 1.

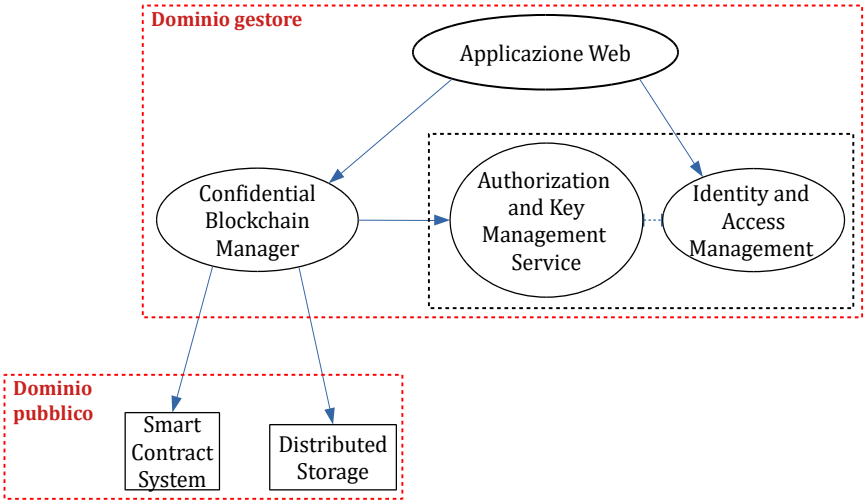


Figura 1 Architettura di alto livello della soluzione progettata

L'architettura ricalca il modello di alto livello descritto nel documento D2.2 "Specifiche del Sistema SmartChain", e si sviluppa su due domini di gestione: il *dominio pubblico*, composto da componenti liberamente accessibili da chiunque, e il *dominio gestore*, sotto il controllo di una o più autorità che sono in grado di regolare le modalità di accesso e utilizzo da parte di entità esterne.

Nell'architettura proposta, si prevedono due componenti principali che devono essere eseguite nel contesto del dominio pubblico: un componente in grado di implementare una logica di smart contract (*Smart contract system*) e un componente in grado di implementare logiche di conservazione dei dati in modo distribuito (*Distributed Storage*). Lo *smart contract system* ha il compito di controllare la corretta esecuzione di logiche applicative in modo decentralizzato e conservare metadati e dati di piccole dimensioni utili a questo scopo. Visto che le tipiche soluzioni specializzate per realizzare *smart contract* hanno lo svantaggio di richiedere alti costi di mantenimento nell'ambito della conservazione di grandi quantità di dati, si prevede che tutti gli altri dati, di dimensioni più consistenti, (ad esempio, *documentazione* e *certificazioni* delle imprese) vengano memorizzati sul sistema di *storage distribuito*. Si prevede che questi due componenti siano realizzati tramite tecnologie indipendenti specializzate per svolgere in modo ottimizzato i compiti previsti. Inoltre, questi due componenti vengono ricercati nell'ambito di tecnologie già esistenti, e si prevede che le tecnologie che vengono utilizzate nella prima versione del progetto siano *Ethereum* per la parte di *smart contract system* e *IPFS* per la parte di storage distribuito.

Si prevede che nello *smart contract system* verranno memorizzate tutte le informazioni di filiera che devono essere messe a disposizione degli *utenti validatori*. D'altra parte, il sistema nel suo complesso deve introdurre delle misure per rispettare i requisiti in termini di confidenzialità delle informazioni richiesti dalle aziende della filiera. Per questo motivo si introduce la piattaforma software eseguita nell'ambito del *dominio gestore*, che intermedia tutte le richieste di accesso degli utenti che devono accedere a informazioni ritenute confidenziali e mette a disposizione delle interfacce utilizzabili anche da utenti inesperti. Si prevede di realizzare la piattaforma tramite un'architettura distribuita composta di software in parte già esistenti nell'ambito dei progetti open source a disposizione, e in parte minore da sviluppare appositamente all'interno del progetto.

All'interno della piattaforma individuiamo tre macro-componenti:

- L'**Applicazione Web** ha lo scopo di mettere a disposizione delle interfacce utilizzabili dalle imprese per gestire le proprie informazioni e dagli utenti che vogliono utilizzare il sistema nell'ambito delle operazioni di approfondimento e verifica sulle informazioni della filiera;
- Il servizio di **Gestione delle identità e dell'autenticazione degli utenti (Identity and Access Management, o IAM)** ha il compito di conservare e regolare la gestione di tutti gli utenti noti al sistema, e di implementare le funzionalità di Single Sign-On per l'autenticazione degli utenti stessi per conto degli altri componenti del sistema;
- Il servizio di **gestione delle autorizzazioni (Authorization Service, o AS)** e di **gestione delle chiavi (Key Management Service, o KMS)** sono i servizi dedicati alla gestione sicura di tutte le chiavi crittografiche necessarie per la cifratura dei dati su blockchain e sul filesystem distribuito. Il KMS espone funzionalità per effettuare operazioni di cifratura, e si appoggia all'AS per regolare le autorizzazioni e consentire l'esecuzione di queste operazioni in base alle necessità del sistema e ai permessi degli utenti che effettuano le richieste;
- Il **Confidential Blockchain Manager (CBM)** ha lo scopo di intermediare l'accesso ai servizi blockchain da parte dell'Applicazione Web e di applicare tutte le misure di sicurezza sui dati memorizzati sui sistemi blockchain acceduti pubblicamente.

Per spiegare le logiche di funzionamento, si considerano le operazioni che devono essere supportate dal sistema. A questo scopo, di seguito si ricordano i profili utente e le operazioni supportate dal sistema così come definite nel deliverable D2.2 "Specifiche del sistema SmartChain", in cui sono disponibili ulteriori dettagli e considerazioni. Nelle successive sezioni si descrivono dettagli di funzionamento più approfonditi per ciascuna operazione.

I profili utente che possono interagire con il sistema sono:

- *Amministratore*: rappresenta una o più persone che gestiscono il sistema all'interno del *dominio gestore*, ha il ruolo di inizializzare il sistema con le imprese;

- *Impresa*: rappresenta un'impresa nel territorio e deve essere in grado di gestire tutte le opportune operazioni per l'inserimento di informazioni sulla propria attività e su quelle dei fornitori. Questo profilo utente deve essere registrato sulla piattaforma da parte di un amministratore;
- *Utente*: rappresenta una qualsiasi entità che vuole effettuare operazioni di verificare di informazioni sul sistema. Questo profilo utente può non essere registrato sulla piattaforma per effettuare le operazioni di verifica rivolte al pubblico.

Nel sistema è inoltre presente un'ulteriore entità identificata come *fornitore esterno*, che ha il ruolo di modellare imprese non registrate alla piattaforma ma di cui vogliamo comunque mantenere informazioni e relazioni con imprese registrate nel sistema.

Le operazioni messe a disposizione dal sistema sono:

- ***R1**: Registrazione di una nuova impresa*
  - Può essere eseguito solo dall'amministratore
- ***R2**: Inserimento di un fornitore esterno a un'impresa*
  - Può essere eseguito solo dall'impresa a cui viene associato il fornitore stesso
- ***R3:** Inserimento di un fornitore interno a un'impresa*
  - Può essere eseguito solo dall'impresa a cui viene associato il fornitore stesso;
- ***R4**: Creazione di una certificazione per la propria impresa*
  - Può essere eseguito solo dall'impresa a cui viene associata la certificazione
- ***R5:** Dichiarazione di una certificazione per un fornitore esterno della mia impresa*
  - Può essere eseguito dall'impresa a cui quel fornitore esterno è associato
- ***R6**: Accesso ai dati di una risorsa gestita dal sistema*
  - La risorsa può essere acceduta secondo un principio di *proprietà (ownership)* della risorsa (ad esempio, le certificazioni delle imprese possono essere sempre accedute dalle imprese a cui sono associate) o di *delega* dell'accesso dal proprietario a un'altra entità.

Viste le funzionalità di alto livello appena elencate e i requisiti di accesso, si introducono in questa fase di progettazione le seguenti ulteriori funzionalità:

- ***R7**: Delega all'accesso di una risorsa*: un'impresa concede l'accesso a una o più informazioni sotto il suo controllo. L'entità delegata può essere sia un'impresa sia un'utente.
- ***R8**: Associazione di una nuova impresa registrata a un fornitore esterno*: l'operazione ha l'obiettivo di modellare la situazione nella quale un'impresa non è registrata a sistema, ma che di fatto ha partecipato come fornitore all'interno della rete di fornitura gestito dal sistema. L'operazione serve a ricondurre le informazioni associate a quel fornitore esterno al nuovo profilo di impresa registrata senza dover reintrodurre tutte le informazioni necessarie.

Si osserva che in questa prima fase di progettazione di alto livello non si dettagliano funzionalità operative di aggiornamento o rimozione di informazioni inserite, su cui ci si occuperà nella fase di progetto di dettaglio.

3.1. Modalità di protezione dei dati

Il sistema ha l'obiettivo di garantire l'accesso selettivo a determinate informazioni, per questo alcuni dati vengono omessi completamente dalle piattaforme di accesso pubblico e vengono mantenuti solo nell'applicazione Web per l'accesso alle sole persone autorizzate. Inoltre, il sistema applica delle strategie di cifratura su tutti i dati accessibili memorizzati nel sistema di smart contract e nello storage distribuito, che sono accessibili al pubblico. La cifratura delle informazioni deve essere applicata considerando due vincoli fondamentali del sistema:

- gli smart contract devono essere in grado di applicare logiche di autorizzazione sui dati degli utenti anche in presenza di alcuni campi cifrati;
- i dati cifrati devono essere potenzialmente controllabili da delle entità esterne che vogliono controllare la correttezza delle informazioni memorizzate.

Di seguito si discute il livello di protezione da applicare e sui dati identificati nella fase di modellazione e definizione delle specifiche (vedere deliverable D2.2).

Dati Impresa

- Identità legale (es: P.IVA)
    - Memorizzati su Smart Contract in modo cifrato
- Ragione Sociale
    - Memorizzati su Smart Contract in modo cifrato
- PEC
    - Memorizzato in database interno
- Nome della persona che ha chiesto la registrazione (esempio: titolare dell'impresa)
    - Memorizzato in database interno
- Documentazione identità: documentazione associata all'impresa (es. pdf della carta di identità)
    - Memorizzato in database interno
- Sede legale (esempio: indirizzo)
    - Memorizzati su Smart Contract in modo cifrato
- Certificazioni: lista delle certificazioni in possesso dell'impresa
    - Memorizzati su Smart Contract in modo parzialmente cifrato (compare la lista degli identificativi opachi e potenzialmente informazioni di alto livello riguardo ciascuna certificazione)
- Documenti di certificazione

    - Memorizzati su file system distribuito in modo cifrato
- Lista dei rapporti di fornitura
    - Memorizzati su Smart Contract in modo parzialmente cifrato (compare la lista degli identificativi opachi e potenzialmente informazioni di alto livello riguardo ciascun rapporto di fornitura)
- Altre informazioni di profilo critiche relative agli anni (esempio: fatturato, numero di ordini, volume merci in ingresso/uscita, volume produzione annuo, numero dei dipendenti, ammontare dei pagamenti ai dipendenti, ecc.)
    - Memorizzate su database interno

Fornitore (impresa non registrata)

- Identità legale: identificatore legale dell'impresa (es. P. IVA)
    - Memorizzata su Smart Contract in modo cifrato
- Nome impresa
    - Memorizzati su Smart Contract in modo cifrato
- Ruolo
    - Memorizzata su Smart Contract in chiaro
- Distretto: inquadra il luogo in modo lasco
    - Memorizzati su Smart Contract in modo cifrato
- Data di validità del rapporto di fornitura
    - Memorizzata su Smart Contract in chiaro

Certificazione

- Descrizione: descrizione informale del contenuto della certificazione
    - Memorizzato su database interno
- Certificatore: identificatore di chi rilascia il certificato
    - Memorizzato su Smart Contract in chiaro
- Tipo: tipo della certificazione
    - Memorizzato su Smart Contract in chiaro
- Oggetto: oggetto della certificazione (organizzazione, processo, prodotto), eventualmente con diciture gerarchiche
    - Memorizzato su Smart Contract in chiaro
- Soggetto: impresa certificata
    - Memorizzato su Smart Contract in modo offuscato (compare id opaco dell'impresa associata alla certificazione)
- Validità: periodo temporale di validità della certificazione
    - Memorizzato su Smart Contract in chiaro

3.2. Funzionalità applicazione Web

L'applicazione Web ha il compito di fornire interfacce adeguate agli utenti del sistema, per consentire un accesso semplice e pratico alle funzionalità disponibili. A questo scopo l'applicazione fornisce un livello di astrazione più alto, e comunica con il componente Identity and Access Management, o IAM, per le operazioni di registrazione e autenticazione degli utenti, e col Confidential Blockchain Manager (CBM), per l'invocazione delle funzionalità del sistema.

L'applicazione web avrà anche il compito di mantenere una serie di informazioni non critiche di profilo relativamente alle imprese registrate (ad esempio informazioni su indirizzo legale, indirizzi e sedi degli stabilimenti operativi, numeri di telefono e dati di contatto, email, persona di riferimento per l'impresa, breve descrizione informativa, indirizzo del sito web ufficiale), appoggiandosi ad un suo DB interno che sarà specifico per l'applicazione. Nella progettazione del DB non sono emersi vincoli particolari, anche perché non presenta criticità in termini di prestazioni o dimensione dei dati.

Attraverso l'applicazione web, un utente/impresa può fare le seguenti operazioni.

Effettuare operazioni di richiesta di registrazione sulla piattaforma

Le imprese si registrano (caricando le informazioni e creando un profilo) e poi vengono certificati da utenti specifici (ad esempio il gestore della piattaforma o un soggetto terzo che certifica le identità), che controllano le informazioni caricate e abilitano il profilo a seguito di una verifica delle identità e delle informazioni che descrivono l'azienda che si vuole registrare. La prima registrazione comprende un set minimo di informazioni identificative. Le fasi quindi sostanzialmente due:

- Registrazione impresa: una impresa inserisce dati come Ragione Sociale, partita IVA, PEC, nome della persona che ha richiesto la registrazione (deve essere il titolare o un suo rappresentante), suo documento identità (con anche documenti PDF di carta d'identità). La registrazione rimane pendente**.**
- Approvazione della richiesta. L'operatore verifica le nuove richieste, controlla i documenti, e abilita la richiesta di profilo sulla piattaforma.

Gestione del profilo

Ogni impresa deve poter aggiornare le informazioni del proprio profilo (contatti, telefono ecc.), che possono cambiare nel tempo.

La gestione del profilo comprende operazioni (inserimento/ modifica) su dati non critici, ad esempio:

- Informazioni descrittive: indirizzo legale, indirizzi e sedi degli stabilimenti operativi, numeri di telefono, email, persona di riferimento per l'impresa, breve descrizione, indirizzo web,

- Oltre alle informazioni descrittive abbiamo un altro set di informazioni, critiche e tracciabili, che possono risultare utili in sede di gestione e controllo della tracciabilità, ad esempio:
- Fatturato, numero di ordini, volume merci in ingresso/uscita, volume produzione annuo, numero dei dipendenti, ammontare dei pagamenti ai dipendenti, ecc.

Registrazione dei propri certificati

Ogni impresa può associare al proprio identificativo delle certificazioni o della documentazione. Sul DB interno sono salvate informazioni descrittive (data del caricamento, validità del documento, oggetto certificato, tipo di certificazione, ecc.). Tramite interfaccia, è possibile gestire i diritti di accesso a tali documenti e indicare i soggetti che possono vedere le varie informazioni.

Registrazione/dichiarazione dei propri fornitori

Tramite l'applicazione, un'impresa registrata può dichiarare i propri fornitori con cui lavora. Inoltre, per ogni fornitore, l'impresa può dichiarare se possiede o meno delle certificazioni.

Gestione delle visibilità verso l'esterno dei dati/certificati caricati

Per ogni informazione inserita, un'impresa deve poterne gestire la visibilità e l'accesso ad altri utenti. Quindi un'ulteriore funzionalità dell'applicazione web è data dalla possibilità di modificare o aggiornare le indicazioni sull'accesso alle informazioni.

Altra funzionalità dell'applicazione web riguarda l'inserimento di interfacce per presentare indicatori riassuntivi dei dati inseriti e resi disponibili dalle imprese. In particolare, l'applicazione web fornisce delle interfacce per visualizzare ad esempio:

- il livello di completezza della descrizione del proprio profilo di un'impresa. È un indicatore automatico - privato per ogni impresa registrata e non visibile quindi da altri;
- un confronto tra il proprio profilo con quelli di altre imprese registrate, per poter vedere come ci si posiziona rispetto agli altri. È un indicatore automatico - privato per ogni impresa registrata e non visibile quindi da altri;
- un insieme di indicatori aggregati pubblici che diano informazioni sull'insieme delle imprese che sono registrate;
- un insieme di indicatori per evidenziare in maniera pubblica i casi più virtuosi del distretto, ad esempio per indicare chi ha il profilo più completo fra tutti.

3.3. Funzionalità smart contract

Le piattaforme di smart contract e storage distribuito hanno lo scopo di offrire informazioni disponibili pubblicamente. Inoltre, i dati conservati sugli smart contract possono essere manipolati solo in base ad interfacce definite a tempo di inizializzazione del sistema. Si ricorda che le interfacce per la lettura dei dati su di uno smart contract sono solo astrazioni applicative e che non impediscono l'accesso anche ad altre informazioni presenti sulle strutture dati dell'infrastruttura blockchain. Le funzionalità di protezione dei dati sono implementate tramite cifratura, come descritto nella Sezione 3.1.

Interfacce per la manipolazione dei dati delle imprese

- **Registrazione impresa:** crea una nuova impresa sullo smart contract sulla base della sua partita IVA (*VATNumber)*, del suo nome (*Name)* e del luogo in cui opera (*Location)*. La funzione restituisce un identificativo opaco univoco per gestire l'impresa all'interno dello smart contract (*EnterpriseId)*.
- **Modifica Impresa**: modifica i dati associati a un'impresa esistente dato il suo identificativo opaco (*EnterpriseId*). Tutte le informazioni associate all'impresa possono essere modificate (*VATNumber, Name, Location)*.
- **Disabilitazione Impresa**: disabilita le funzionalità legate a un'impresa sullo smart contract. Richiede come input l'identificativo opaco dell'impresa stessa (*EnterpriseId)*. Si osserva che, data la natura immutabile dei dati storici conservati all'interno dello smart contract, non è possibile realizzare una funzione "elimina". Eventualmente, nel caso di dati protetti da sistemi di cifratura (si veda la Sezione 3.1), sarà possibile eliminare completamente le chiavi di cifratura conservate nel gestore di chiavi per impedire futuri accessi ai dati presenti all'interno dei servizi blockchain pubblici.
- **Inserimento Fornitore Registrato:** dati gli identificativi opachi di due imprese registrate sullo smart contract (*CustomerEnterpriseId* e *SupplierEnterpriseId*), si crea un rapporto di fornitura tra le due imprese. Questo rapporto di fornitura può essere caratterizzato da due informazioni: una stringa descrittiva del ruolo del rapporto di fornitura (*Role*) e la data che indica la fine del rapporto di fornitura (*Validity*). Si nota che la data di inizio della fornitura è implicita, perché se l'informazione di fornitura è presente nello smart contract e se la data di fine del rapporto è "nel futuro", vuol dire che la fornitura è attualmente valida. Questa funzionalità restituisce un identificativo opaco che identifica il rapporto di fornitura appena inserito (*RelationId*).
- **Inserimento Fornitore Non Registrato**: se un'impresa vuole aggiungere informazioni su fornitori attualmente non presenti nello smart contract, non può invocare la precedente funzione di *Inserimento Fornitore Registrato* perché non può inserire alcun identificativo opaco per il fornitore (il precedente campo *SupplierEnterpriseId*). Per questo motivo, questa funzionalità richiede dall'impresa di indicare atomicamente

informazioni dell'impresa fornitrice simili a quelle che solitamente vengono inserite nella registrazione di un'impresa. In particolare, la partita iva dell'impresa fornitrice (*VATNumber*), il nome (*Name*), il luogo in cui opera (*Location*). Infine, sono richiesti le medesime informazioni riguardanti il tipo di rapporto di fornitura (*Role*) e la data di fine rapporto (*Validity)*. Questa funzionalità restituisce un identificativo opaco che identifica il rapporto di fornitura appena inserito (*RelationId*).

- **Modifica Rapporto di Fornitura**: dato l'identificativo del rapporto di fornitura (*RelationId*), permette di modificare le informazioni riguardanti il rapporto di fornitura (*Role*) e la data di fine rapporto (*Validify*).
- **Modifica Rapporto di Fornitura con Fornitore non Registrato**: dato l'identificativo del rapport (*RelationId*), permette di modificare la partita iva (*VATNumber*), il nome (*Name*) e il luogo in cui opera (*Location*) dell'impresa fornitrice, oltre alle informazioni riguardanti il rapporto di fornitura (*Role*) e la data di fine rapporto (*Validify*).
- **Invalidazione di Rapporto di Fornitura**: dato l'identificativo del rapport (*RelationId*), rimuove il rapporto di fornitura associato a un'impresa committente.

Interfacce per la manipolazione delle certificazioni

- **Inserimento Certificazione**: inserisce una nuova certificazione associandola a un'impresa. L'impresa è indicata tramite il suo identificatore opaco (*EnterpriseId*), mentre le informazioni che caratterizzano la certificazione sono il nome dell'emettitore del certificato (*Issuer*), il tipo di certificato (*CertType*), l'oggetto della certificazione (*CertificationObject*), la data di scadenza della certificazione (*Validity*). Questa interfaccia genera e restituisce l'identificativo della certificazione (*CertificationId*).
- **Modifica Certificazione**: modifica una certificazione esistente dato il suo identificativo (*CertificationID*). Tutti i campi del certificato possono essere oggetti di modifica: il nome dell'emettitore del certificato (*Issuer*), il tipo di certificato (*CertType*), l'oggetto della certificazione (*CertificationObject*), la data di scadenza della certificazione (*Validity*).
- **Eliminazione Certificazione**: rimuove l'associazione di una certificazione esistente da un'azienda dato il suo identificativo (*CertificationID*).

Interfacce standard per l'accesso ai dati

- **Informazioni Impresa**: dato l'identificativo opaco di un'impresa (*EnterpriseId*), restituisce le informazioni associate.
- **Ispezione Permessi di Autorizzazione**: dato l'identificativo di un'impresa (*EnterpriseId*), di una relazione (RelationId) o di una certificazione (*CertificationId*), permette a chi invoca la funzione di sapere se ha i permessi per effettuare operazioni di modifica dei dati su sull'oggetto associato all'identificatore. Questa funzione è utilizza soprattutto

dalle altre funzioni implementate dallo smart contract, ma viene esposto per essere utilizzabile anche da altri componenti e per restituire informazioni riguardanti i permessi di modifica a ciascuna impresa.

- **Elenco Certificazioni**: dato l'identificativo opaco di un'impresa (*EnterpriseId*), restituisce l'elenco degli identificativi delle certificazioni associate come fornitori.
- **Lettura Certificazione**: dato l'identificativo opaco di una certificazione (*CertificationId*), restituisce le informazioni della certificazione associata.
- **Lettura Rapporto di Fornitura**: dato l'identificativo opaco di una fornitura (*RelationId*), restituisce le informazioni di fornitura associate.

3.4. Gestione delle autorizzazioni e delle chiavi

Il sistema progettato istanzia logiche di gestione delle autorizzazioni a molteplici livelli. In questa fase di progetto, ci interessa soprattutto approfondire le logiche di autorizzazione applicate in modo integrato dalle funzioni implementate dallo Smart Contract e dal KMS. Concettualmente, le due logiche di autorizzazione hanno un ruolo duale: le prime (smart contract) hanno il ruolo di limitare la scrittura di informazioni rese pubblicamente disponibili; le seconde (KMS) hanno il ruolo di limitare l'accesso e la lettura alle informazioni soltanto agli attori autorizzati. La direzione del progetto è quella di poter implementare una logica di autorizzazione sui dati cifrati in maniera simile a come si può applicare sui dati in chiaro, ovvero in modo analogo a come è stato definito in fase di definizione di specifiche. Intuitivamente, questo è possibile perché la cifratura viene applicata su dati sulla base dei quali gli smart contract non applicheranno logiche di autorizzazione (ad esempio, informazioni associate alle imprese, certificazioni). In fase di progettazione di dettaglio verrà spiegata in modo più approfondito l'implementazione di queste logiche, mostrando come effettivamente queste non sono limitate dai dati cifrati. In questa fase si è quindi più interessati ad approfondire alcuni aspetti della progettazione delle chiavi di cifratura e di autorizzazione del KMS.

Idealmente, il KMS permette di associare una chiave di cifratura per ogni risorsa memorizzata su smart contract, in modo da gestire in modo flessibile e indipendente la cifratura di tutte le informazioni. La necessità di impiegare più chiavi deriva da molteplici motivi:

- si vuole rendere possibile rilasciare selettivamente chiavi di cifratura specifiche nel momento in cui si vuole delegare l'accesso a specifici dati da parte di altre imprese o di utenti esterni, e si vogliono consentire operazioni di verifica sulla veridicità dei dati inseriti sul backend blockchain da parte specifiche imprese o utenti registrati al sistema;
- in versioni più avanzate, si ipotizza di voler essere in grado di ruotare efficientemente solo alcune chiavi.

Il sistema deve essere in grado di gestire autorizzazioni per capire quali utenti possono effettuare queste operazioni. Idealmente, le scelte che guidano se concedere le autorizzazioni si basano su molteplici fattori:

- sulla base del ruolo, come nel caso di Admin, Imprese o Utenti;
- sulla base delle proprietà di accesso associate a ciascuna specifica risorsa ("capabilities"), sia nell'ambito dei "proprietari" di ciascuna risorsa (ogni impresa è "proprietaria" e responsabile dei dati dell'impresa caricati, come ad esempio in un filesystem) sia di utenti che possono essere stati delegati ad accedere a quei dati dall'impresa proprietaria. Nella abella 1 si enumerano alcune delle operazioni di autorizzazione necessarie;
- Sulla base di regole aggiuntive specificamente definite.

Tabella 1 Tabella dei requisiti di autorizzazione dal punto di vista dei servizi di gestione delle autorizzazioni e gestione chiavi

Operazione	**Autorizzazione (azione/risorsa)**	**Policy**
Registrazione nuova impresa	Crea/Impresa	Solo 'admin' può farlo
Inserimento fornitore esterno	Crea/Fornitore	Impresa può aggiungere solo forniture riguardo sé stessa
Inserimento fornitura impresa interna	Crea/Fornitura	Impresa può aggiungere solo forniture riguardo sé stessa
Creazione certificazione propria	Crea/Certificazione	Impresa può creare una certificazione solo per sé stessa
Dichiarazione certificazione fornitore esterno	Crea/Dichiarazione	Impresa può creare una dichiarazione certificazione solo per fornitori esterni a lei associati
Accesso a dati risorsa	Leggere/<risorsa>	L'entità è stata delegata dall'impresa proprietaria a leggere i dati

3.5. Flussi delle operazioni

3.5.1. Flussi di autenticazione e autorizzazione

Il sistema progettato fa uso di un'architettura decentralizzata con componenti dedicati alla gestione delle identità. Si prevede di impiegare componenti open-source compatibili con i più moderni standard di autenticazione single sign-on e di autorizzazione delegata per una corretta gestione dei token di autenticazione rispetto alle autorizzazioni concesse ai componenti

dell'architettura e degli utenti. Già in fase preliminare di progettazione si pianifica di utilizzare gli attuali standard di riferimento, ovvero OpenID per gestione di autenticazione e OAuth2 per la gestione delle autorizzazioni. Di seguito si propone uno schema di alto livello dei flussi di operazioni che coinvolgono i componenti e i protocolli.

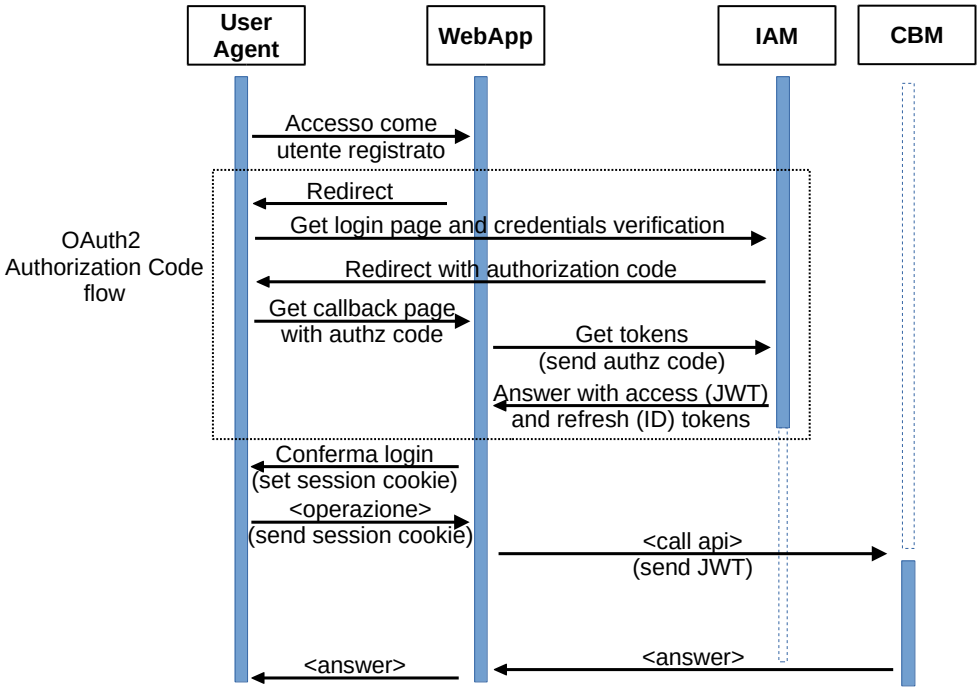


Figura 2 Flusso operazioni per single sign-on e operazioni autorizzate

Per permettere di accedere a funzionalità privilegiate, un utente deve effettuare il login presso la piattaforma. A questo scopo, l'utente impiega uno user agent (browser) e "clicca" sull'apposito pulsante di autenticazione messo a disposizione dalla WebApp. La WebApp non gestisce direttamente il sistema di gestione degli utenti registrati e delle autorizzazioni ad essi associati, ma delega queste funzionalità all'IAM. Per questo motivo il browser deve effettuare il login tramite single sign-on presso l'IAM. A questo scopo, la WebApp inizia un flusso di operazioni standard del protocollo OAuth2, nella modalità di rilascio dei token tramite *Authorization Code flow*. La figura mostra le operazioni di alto livello di questo flusso di operazioni: nel momento in cui lo user agent invia la richiesta HTTP in seguito all'interazione dell'utente con l'interfaccia, la WebApp risponde con codice redirect e indicando come destinazione uno URI apposito gestito dall'IAM. L'IAM, dopo le verifiche del caso riguardo la correttezza della richiesta ricevuta, risponde mostrando all'utente la pagina di login (se l'utente non era già loggato presso l'IAM). L'utente esegue la procedura di autenticazione presso l'IAM (ad esempio, inserendo le credenziali, ma potenzialmente utilizzando anche ulteriori procedure di two-factor authentication, o potenzialmente anche altre moderne modalità di autenticazione), e in caso di successo richiede all'utente la validazione dei permessi di accesso che si vogliono concedere alla WebApp. In seguito alla conferma di queste informazioni, lo user

agent riceve in risposta una risposta HTTP redirect destinata a un apposito URI gestito dalla WebApp (il cosiddetto *redirect_uri*) che include un *authorization code*. Lo user agent "segue" la redirect ed effettua la richiesta presso la WebApp, che potrà scambiare l'*authorizazion code* con i token di autenticazione (*access token* e *refresh token*) effettuando direttamente una richiesta presso l'IAM. In ogni successiva operazione, la WebApp potrà allegare l'access token nelle sue successive operazioni presso il CBM (o eventualmente anche presso il KMS) per richiedere che le operazioni richieste siano gestite in vece dell'utente loggato. Si prevede che l'access token sarà implementato tramite protocollo standard JWT e che concettualmente sia un'attestazione crittografica autenticata tramite una firma digitale. Per validare la legittimità dell'*access token*, ogni servizio dovrà essere a conoscenza della chiave pubblica dell'IAM. A questo scopo, si prevede di impiegare protocolli di *discovery* definiti nello standard OpenID (i servizi che validano la legittimità del JWT ottengono le chiavi pubbliche tramite richieste presso URI speciali gestiti dall'IAM, che permettono il recupero automatico delle chiavi pubbliche necessarie).

3.5.2. Registrazione di un'impresa

La progettazione di alto livello riguardante le operazioni che soddisfano le funzionalità *R1 "Registrazione di un'impresa" e R8 "Inserimento di un fornitore esterno a un'impresa"* è rappresentata in Figura 3.

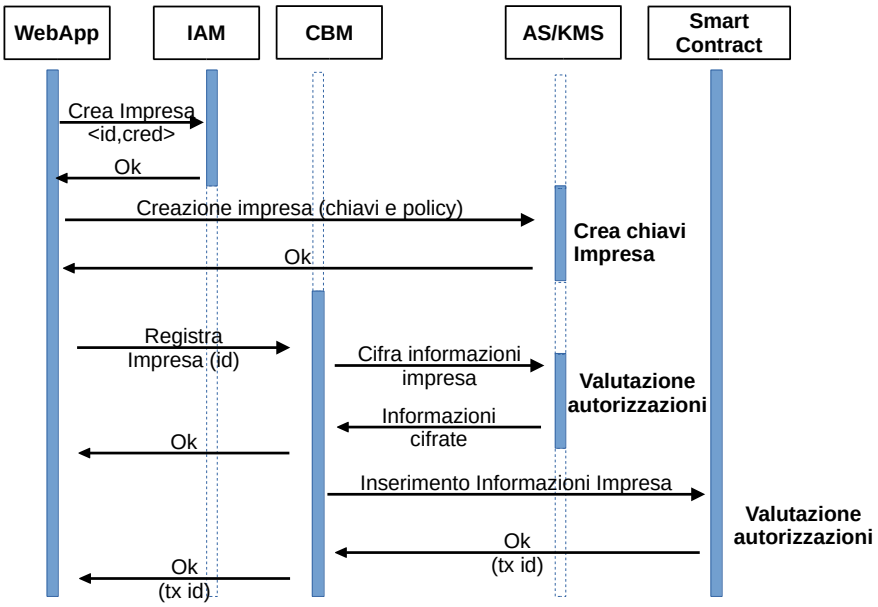


Figura 3 Registrazione di una nuova impresa (si assume che il flusso di operazioni venga eseguito da un utente amministratore del sistema, si omettono informazioni riguardo i token di autenticazione fra i diversi componenti)

La registrazione di un'impresa è un'operazione che può essere effettuata da un utente con privilegi di amministrazione del sistema (si suppone che queste operazioni siano effettuate in

seguito all'esecuzione di un processo di autenticazione come descritto nella sezione precedente, e che la WebApp e l'IAM abbiano validato che l'utente sia associato con i corretti privilegi di accesso). Tramite un'apposita interfaccia grafica, l'utente può inserire tutti i dati necessari sulla WebApp, che internamente memorizzerà tutte le informazioni che caratterizzano l'impresa, di cui non discutiamo i dettagli in questo documento perché possono essere gestiti tramite tradizionali sistemi di gestione delle informazioni. La WebApp deve però comunicare ad altri componenti del sistema la creazione della nuova impresa per consentire la gestione dei dati che verranno memorizzati su infrastrutture blockchain. A questo scopo, la WebApp invoca delle funzionalità apposite dell'IAM per creare un nuovo utente, impostando le informazioni e i privilegi di accesso relative al ruolo di *impresa* all'interno del sistema. La WebApp potrebbe dover invocare delle funzionalità apposite per registrare la nuova impresa presso il KMS, in modo tale che siano create delle chiavi crittografiche e delle policy appropriate per la gestione delle informazioni relative all'impresa stessa.

In seguito alla configurazione dell'IAM e del KMS tramite le apposite API, la WebApp può inserire informazioni di tracciamento dell'impresa presso i servizi blockchain. A questo scopo, invoca delle funzionalità del CBM che riguardano l'inserimento di una nuova impresa, includendo le informazioni necessarie (si veda la Sezione 3.1). Il CBM richiede la cifratura delle informazioni al KMS, che concederà l'esecuzione delle operazioni dopo aver valutato le autorizzazioni associate alla richiesta (in questo caso, si assume che l'amministrazione di sistema abbia la possibilità di modificare le informazioni che caratterizzano l'impresa e quindi il suo token di accesso sia validato correttamente nell'ambito della cifratura delle informazioni associate). Infine, il CBM riceve le informazioni cifrate dal KMS, e può memorizzarle sui servizi blockchain impiegando le apposite funzionalità messe a disposizione dallo smart contract (si veda la Sezione 3.3).

3.5.3. Inserimento di fornitori

Si descrive la progettazione di alto livello dell'operazione che soddisfa la funzionalità *R2 "Inserimento di un fornitore esterno ad un'impresa"* e *R3 "Inserimento di un fornitore interno ad un'impresa"*. Le due funzionalità sono realizzate in modo molto simile, con l'unica differenza che l'inserimento di informazioni relative a un fornitore esterno non richiede l'interazione con l'IAM, perché per definizione dell'operazione stessa il fornitore non è registrato presso il sistema. L'inserimento di informazioni relative a fornitori registrati a sistema richiedono invece che il fornitore in oggetto sia stato precedentemente registrato tramite la funzionalità descritta nella Sezione 3.5.2. La procedura viene rappresentata in Figura 4.

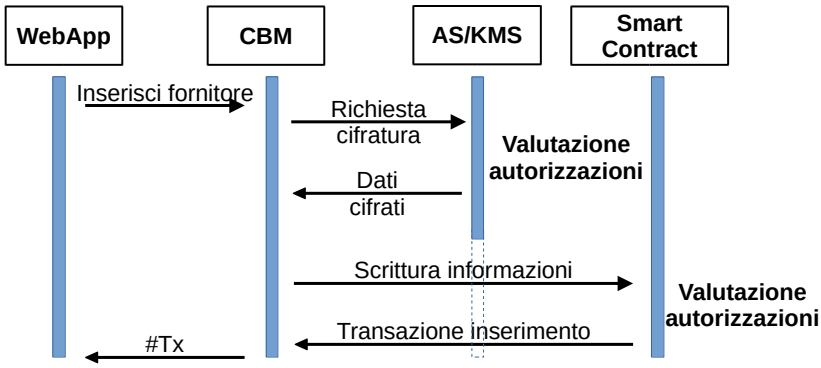


Figura 4 Inserimento fornitore a un'impresa (si assume che il flusso venga eseguito da utente impresa, si omettono informazioni relative ai token di autorizzazione)

L'operazione prevede che, in seguito alla richiesta da parte di un utente loggato a sistema con privilegi di *impresa*, la WebApp invochi un'apposita funzionalità del CBM per inserire un nuovo fornitore alla propria impresa. Il CBM richiede la cifratura delle informazioni protette riguardanti il fornitore al KMS (che, come visto per le altre operazioni, valuta se l'utente ha la possibilità di effettuare questa operazione sulla base delle policy configurate e del token di accesso ricevuto dal CBM). Il CBM riceve i dati cifrati e inserisce le informazioni sullo smart contract.

3.5.4. Inserimento delle certificazioni

Si descrive la progettazione di alto livello dell'operazione che soddisfa le funzionalità *R4* "*Creazione di una certificazione per la propria impresa*" e *R5 "Dichiarazione di una certificazione per un fornitore esterno della mia impresa".*

L'obiettivo dell'operazione è permettere di associare una certificazione a un'impresa all'interno del sistema. L'operazione deve essere consentita solo all'impresa stessa che deve fare l'operazione, e avrà l'effetto di inserire sullo storage distribuito il file attestante la certificazione (ad esempio, il PDF contenente la scansione di un documento, o un file multimediale digitale), e sullo smart contract dei metadati riguardo la certificazione sullo smart contract.

Per descrivere il flusso delle operazioni coinvolte nell'inserimento di una certificazione si fa riferimento alla igura 5 , in cui si descrive la procedura a partire dall'invocazione da parte dell'Applicazione Web (WebApp), assumendo che la WebApp abbia verificato l'identità dell'impresa e ottenuto un token di autenticazione da inviare nelle sue richieste al CBM per attestare l'identità dell'impresa per la quale vengono fatte le richieste (*ovvero implementare il concetto di identity propagation fra i componenti)*. Assumiamo inoltre che la WebApp sia in grado di generare degli identificatori opachi univoci per ogni risorsa gestita dal sistema (*UUID*).

L'obiettivo dell'operazione è di memorizzare i dati e i metadati sui sistemi blockchain. Il sistema deve fare in modo che i dati e i metadati vengano cifrati in maniera opportuna e

memorizzati sul sistema di storage distribuito e sugli smart contract, rispettivamente. La sequenza di operazioni è la seguente:

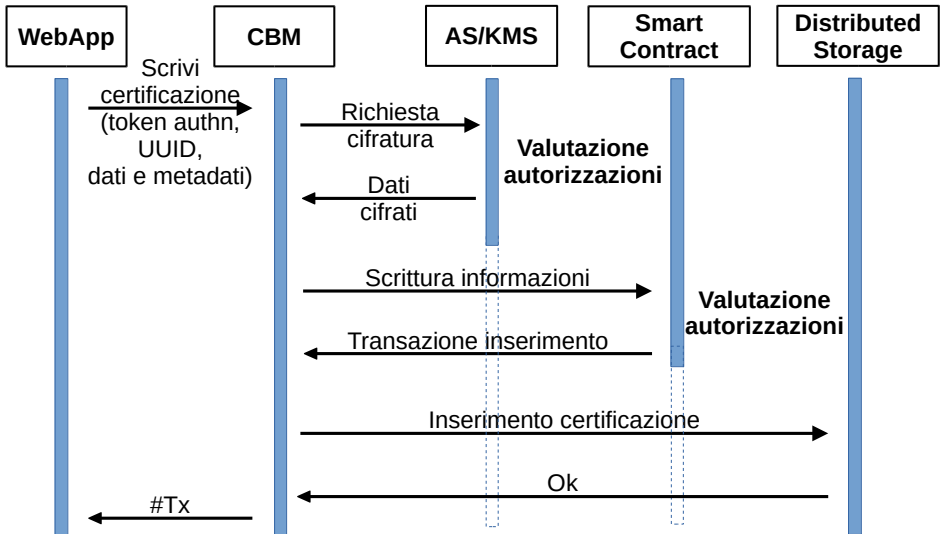


Figura 5 Inserimento di una certificazione di un'impresa (si assume che il flusso di operazioni sia eseguito da un utente di tipo impresa, si omettono informazioni relative ai token di autorizzazione)

- La WebApp definisce l'identificatore opaco per la certificazione (UUID) e invoca la funzionalità di inserimento di una nuova certificazione del Confidential Blockchain Manager (CBM) inviando i dati e i metadati della certificazione, l'identificatore UUID, e il token di autenticazione;
- Il CBM invia la richiesta di cifratura dei dati e dei metadati al KMS inviando i dati e il token di autenticazione al KMS e lo UUID ricevuto dalla WebApp.
- Il KMS effettua la valutazione delle autorizzazioni rispetto al token di autenticazione per validare il ruolo dell'entità che fa la richiesta (la richiesta di inserimento può essere effettuata solo da un'impresa) e rispetto ai dati che devono essere inseriti. Il KMS cifra i dati in modo opportuno e restituisce i dati cifrati al CBM se le autorizzazioni risultano corrette, altrimenti nega l'esecuzione dell'operazione.
- Il CBM esegue uno smart contract per creare l'oggetto sul sistema di smart contract. Lo smart contract di creazione verifica l'autorizzazione a inserire il dato da parte dei CBM, ed effettua dei controlli riguardo per verificare la possibilità di inserire la certificazione da parte dell'impresa.
- Se necessario, il gestore blockchain inserisce anche i dati nel sistema di file system distribuito.

Come ulteriore osservazione preliminare riguardo l'implementazione di queste operazioni, si valuta che viste le caratteristiche tecniche delle tecnologie blockchain pubbliche è altamente probabile che l'operazione di inserimento offerta dal gestore blockchain debba essere esposta

come operazione dal CBM alla WebApp. Si investigheranno più in dettaglio questi aspetti nella fase di progetto di dettaglio

3.5.5. Delega dell'accesso alle informazioni

La funzionalità R7 "*Delega all'accesso di una risorsa*" viene gestita inserendo nuovi account presso il servizio IAM (nel caso in cui l'entità delegata non abbia ancora un account) e aggiungendo nuove policy "ad-hoc" presso il KMS per concedere la decifratura di informazioni in modo selettivo. L'applicazione Web si preoccupa di istanziare interfacce utente user-friendly per consentire l'utilizzo di queste funzionalità in modo trasparente e facile da parte di personale delle imprese. Il flusso delle operazioni viene rappresentato in Figura 6.

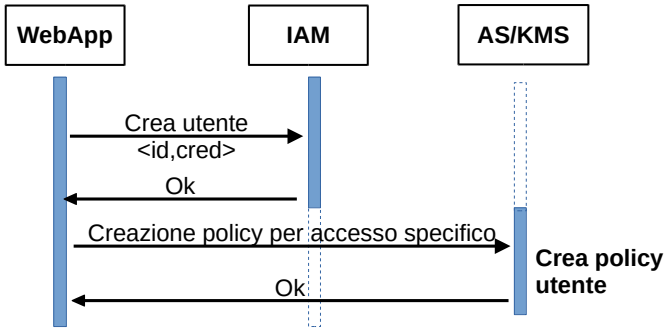


Figura 6 Flusso operazioni per la delega all'accesso di informazioni

Questa funzionalità prevede che la WebApp crei un utente apposito presso l'IAM (associando permessi di accesso comparabili ad un utente *guest*), e configuri policy di accesso presso il KMS specifiche per l'accesso alle informazioni di una o più imprese. Chiaramente, per successive deleghe di accesso allo stesso utente non è richiesta la creazione di un account presso l'IAM. Successivamente, l'utente può utilizzare la funzionalità di lettura delle informazioni per accedere ai dati a cui è stato delegato l'accesso (vedi Sezione 3.5.6).

3.5.6. Lettura di informazioni delle imprese

Si descrive la progettazione di alto livello delle operazioni per soddisfare le funzionalità *R6* "*Accesso ai dati di una risorsa gestita dal sistema*".

L'obiettivo di questa operazione è leggere informazioni riguardanti una certificazione precedentemente inserite nel sistema. Descriviamo questa operazione come potenzialmente eseguibile sia da Imprese sia da Utenti. Come definito in Sezione 0, la certificazione è identificata da un identificativo opaco (UUID), è memorizzata nel sistema di storage distribuito, ed è associata a metadati potenzialmente cifrati mantenuti nel sistema di Smart Contract.

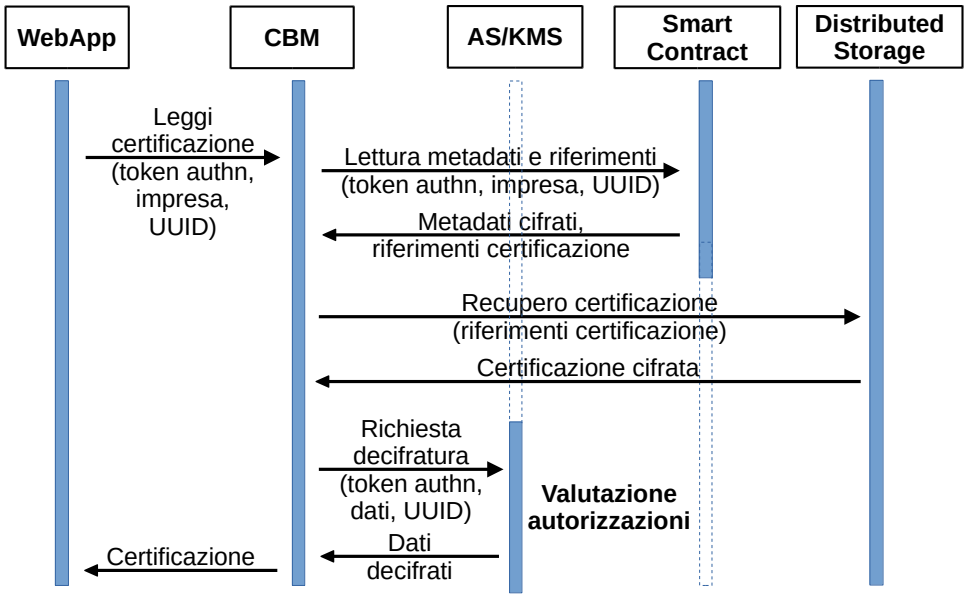


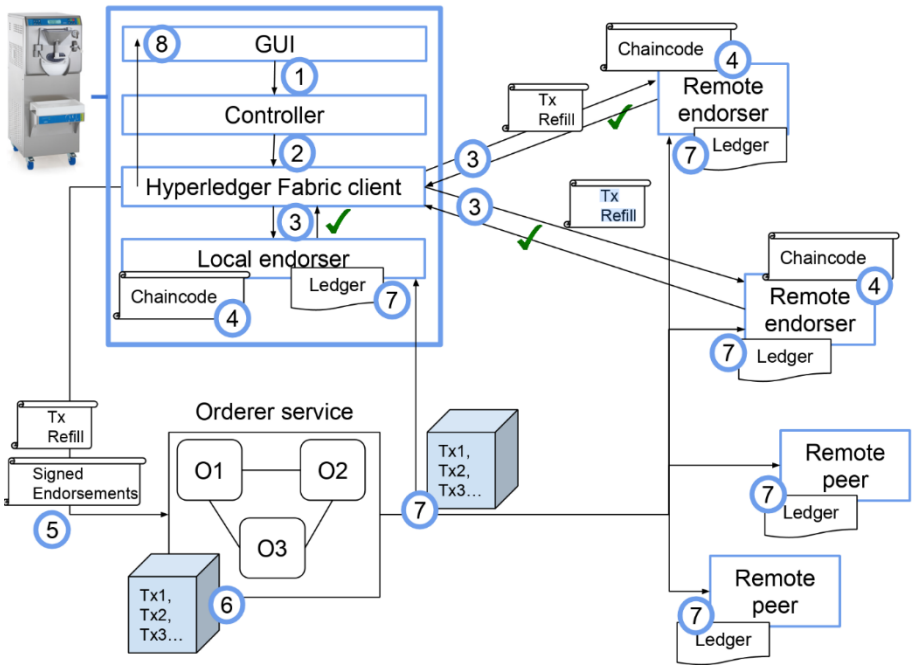
Figura 7 Lettura di una certificazione di un'impresa

La sequenza di operazioni di alto livello eseguite dall'architettura è la seguente:

- La WebApp invoca la funzionalità messa a disposizione dal CBM per ottenere una certificazione inviando il token di autenticazione dell'utente per il quale si sta effettuando la richiesta, lo UUID della certificazione richiesta, l'impresa associata alla certificazione.
- Il CBM contatta lo Smart Contract per leggere le informazioni associate all'impresa e alla certificazione identificata dallo UUID ricevuto dalla WebApp, ottenendo metadati cifrati e i riferimenti riguardanti il mantenimento dei dati della certificazione dello storage distribuito. Il CBM contatta quindi lo storage distribuito e recupera la certificazione cifrata.
- Il CBM effettua al KMS una richiesta di decifratura per i dati e i metadati ricevuti, inviando il token di autenticazione, i metadati e lo UUID della certificazione.
- Il KMS valuta le autorizzazioni dell'entità che effettua la richiesta (impresa proprietaria dell'autorizzazione, o impresa o utente delegati all'accesso della certificazione), e se in caso di autorizzazione valida decifra le informazioni e le restituisce al CBM.
- Il CBM restituisce i dati e i metadati decifrati alla WebApp.

4. Progettazione sistema Carpigiani

Partendo dai componenti di Hyperledger Fabric, è stata individuata una infrastruttura che permette di fare interagire il client con il controller della macchina di gelato di Carpigiani. Nella Figura qui sotto vengono illustrati i componenti e gli step principali di tale infrastruttura.



L'entry point della soluzione individuata è un'interfaccia grafica che lega il sistema implementato da Carpigiani con la rete Fabric (step 1). Un controller realizzato da Carpigiani riceve le informazioni di refill dalla macchina da gelato e invia i dati al client Fabric (step 2) che, a sua volta, genera una proposta di transazione e la invia agli endorser del canale (step 3). Ogni organizzazione, per partecipare alla fase di approvazione (step 3), possiede almeno un peer endorser per ogni canale al quale partecipa. Questo dà la possibilità a tutte le organizzazioni di partecipare alla fase di approvazione garantendo la fiducia tra organizzazioni che possono avere l'interesse di frodare gli altri partecipanti (ad esempio se un cliente si rifornisse da un produttore diverso da quello concordato). Sempre per lo stesso fine, è stata scelta una politica di approvazione (fase di endorsement) del tipo AND, cioè tutte le organizzazioni partecipanti al canale, devono approvare la proposta di transazione firmando (step 4). Firmare l'approvazione, è una ulteriore garanzia della proprietà di fiducia e non ripudio.

Se tutti gli endorser approvano la transazione proposta dal client che si interfaccia con la macchina, allora questo invia la transazione agli orderer (step 5).

Fabric consente di implementare uno tra queste tre tipologie di orderer:

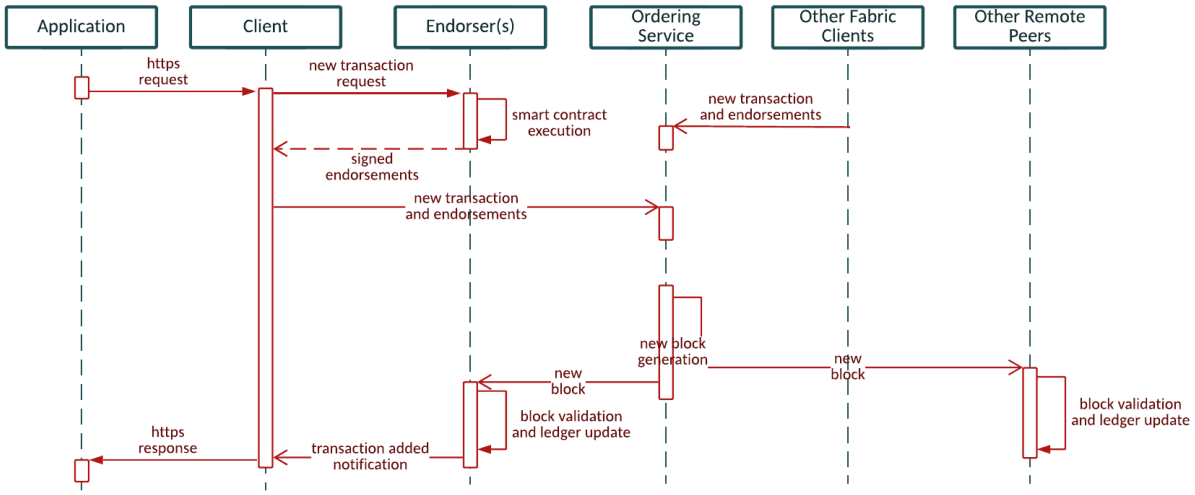
- **Solo**, si tratta di un servizio di ordering costituito da un singolo nodo orderer che pertanto non garantisce tolleranza ai guasti. Viene utilizzata unicamente in quelle situazioni che vedono lo svolgimento di semplici test.
- **Kafka**, è un'implementazione Crash Fault Tolerant (CFT) che utilizza una configurazione dei nodi di tipo "leader e follower". Kafka utilizza un insieme ZooKeeper per scopi di gestione. Il servizio di ordinazione basato su Kafka è basato su un cluster di orderer della stessa organizzazione che partecipano al voto. Questa struttura logicamente centralizzata e tecnologicamente complessa può portare ad un sovraccarico amministrativo aggiuntivo da parte dell'organizzazione eletta gestore del cluster Kafka.
- **Raft**, è un servizio di ordinazione CFT basato su cluster di orderer in cui i nodi possono appartenere a organizzazioni differenti. Raft vede l'implementazione dell'omonimo protocollo che segue un modello "leader e follower", in cui per ogni canale viene eletto un nodo leader fra il pool di orderer e le sue decisioni vengono replicate dai follower. Il servizio di ordering di Raft è più facile da configurare e gestire rispetto ai servizi di ordering basato su Kafka ed è stato progettato appositamente per consentire a diverse organizzazioni di contribuire con uno o più nodi al servizio di ordering distribuito.

Per fornire una politica di consenso il meno centralizzata possibile, è stata scelta la tipologia Raft che garantisce resilienza e decentralizzazione. Inoltre permette di suddividere e distribuire differenti gruppi di orderer tra la rete migliorando la gestione e velocità della piattaforma. I nodi orderer, dato che partecipano a tutta la rete, possono ricevere diverse transazioni inerenti a canali differenti e considerato che in media vengono effettuati due/tre refill (transazioni) al giorno, si è scelto come partenza un numero di cinque nodi (che potrebbe aumentare in caso di carico eccessivo) per la gestione dell'intera rete. Per la scelta del numero di orderer iniziali, si è considerato il trade-off tra resilienza (più orderer sono presenti e più la rete è resiliente) e velocità di creazione dei blocchi (più orderer partecipano al voto, più scambi di messaggi dovranno essere computati prima di raggiungere il quorum).

Tutti gli orderer appartenenti al servizio di ordering creano un nuovo blocco nel quale inseriscono le transazioni che fanno riferimento a uno specifico canale (step 6). Questo blocco sarà poi invitato in broadcast a tutti i peer appartenenti al canale e sarà quindi ricevuto da tutte e tre le organizzazioni che vi partecipano (step 7). Una volta che tutti i peer del canale hanno ricevuto il blocco, questo viene validato (verificano che il flusso delle transazioni sia valido) e successivamente ne effettuano il commit (lo inseriscono) nel ledger (step 7). Più committer partecipano al canale, più nodi detengono il ledger aumentando la resistenza alla manomissione della Blockchain e, quindi, la sicurezza di tutta l'infrastruttura. Per questo motivo è stato scelto di realizzare più peer per ogni organizzazione. Un attore che partecipa a molti canali avrà endorser e committer che parteciperanno a più terne così da migliorare il load balancing e la velocità in fase di endorsement, oltre che la resilienza e la sicurezza. Una volta effettuati il commit del blocco, il client riceve il messaggio che la transazione è stata aggiunta

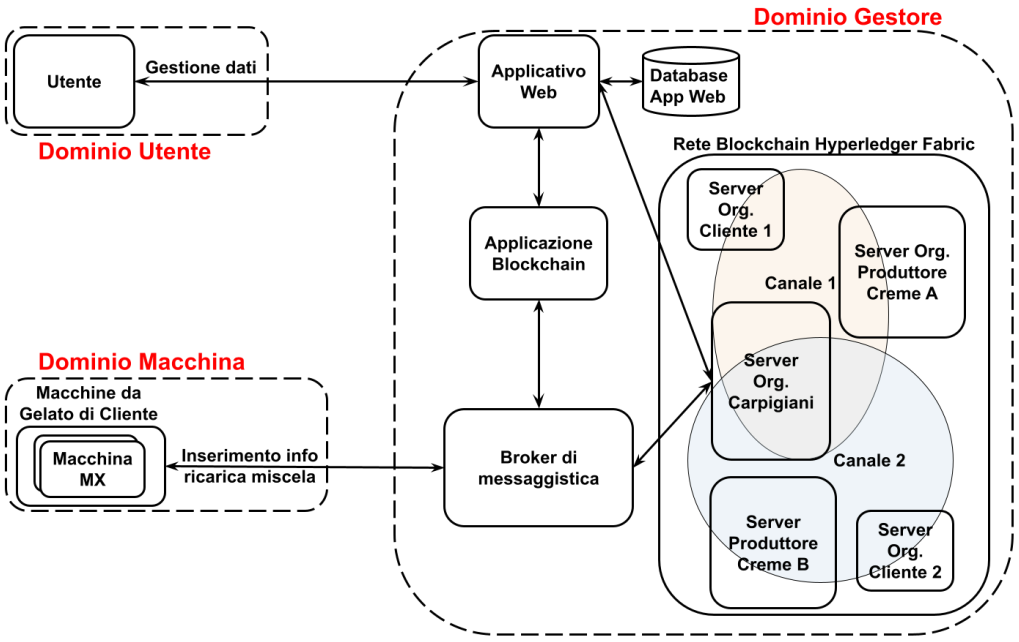
alla Blockchain con successo o dell'eventuale errore. Dopodiché inoltra il messaggio alla GUI di Carpigiani dalla quale è possibile visualizzarne il contenuto (step 8).

Nella figura sotto è riportato anche il diagramma di sequenza delle procedure per l'inserimento nella blockchain delle informazioni di refill effettuate presso una macchina da gelato.



4.1. Domini applicativi e componenti

Entrando più nel dettaglio della progettazione dell'architettura dal punto di vista dei domini applicativi, il sistema è stato progettato seguendo tre classi di dominio (mostrate graficamente nella figura sotto).



Nello specifico le classi di dominio si suddividono in dominio utente, dominio gestore e dominio macchina.

Il **dominio utente**, ha il compito di gestire le informazioni registrate nella blockchain accedendo ad un applicativo Web, vede coinvolti il personale tecnico di Carpigiani (in qualità di amministratore) e il personale non tecnico delle aziende coinvolte nella raccolta delle informazioni di refill (in qualità di utenti).

Il **dominio gestore** ha il compito di gestire la rete Blockchain fornendo anche gli strumenti esterni per la comunicazione con essa in lettura e scrittura dei dati. Questo è costituito in particolare da:

- **Rete Blockchain**, è il nucleo principale dell'architettura e vede coinvolti i componenti descritti in precedenza (client, endorser, commiter e orderer), i componenti possono essere suddivisi su più server afferenti alle diverse organizzazioni partecipanti alla Blockchain. La rete è logicamente suddivisa in canali isolati gli uni dagli altri. Per tale componente si è pensato di utilizzare il software Hyperledger Fabric.
- **Broker di messaggistica**, è utilizzato principalmente per disaccoppiare e gestire in modo concorrente le richieste di inserimento dei dati di refill provenienti dalle macchine da gelato e le richieste di modifica o gestione dei canali della Rete Blockchain. Per tale componente si è pensato di utilizzare il software RabbitMQ.
- **Applicativo Web**, ha il compito principale di presentare i dati relativi ai refill delle macchine da gelato, interrogando la Blockchain e fornendo anche strumenti aggiuntivi utili per un utente quali strumenti di ricerca fra differenti macchine afferenti ad un unico proprietario. Nel caso di un utente admin, fornisce anche funzionalità di creazione di account per utenti dell'applicazione web e, con l'ausilio dell'**Applicazione Blockchain**, fornisce le funzionalità di base per la gestione dell'architettura della Blockchain (aggiunta di una macchina da gelato, aggiunta di una nuova organizzazione, aggiunta di un nuovo canale e installazione di un nuovo smart contract nel canale). Per l'implementazione di tale componente si è pensato di utilizzare il framework Ruby on Rails.

Al dominio gestore partecipa unicamente il personale tecnico di Carpigiani.

Il **dominio macchina** è di fondamentale importanza in quanto è deputato all'inserimento delle informazioni di refill che sono le informazioni sulle quali si basa l'intera architettura progettata. L'inserimento delle informazioni avviene secondo lo schema descritto nei paragrafi precedenti e l'attore coinvolto è rappresentato da un software in esecuzione sulle macchine da gelato.

4.2. Funzionalità smart contract

Il sistema progettato ha l'obiettivo di salvare all'interno della Blockchain le informazioni relative ai refill delle macchine da gelato in modo che tutti i partecipanti ai canali possano approvarle e che, in un secondo momento, non sia possibile ripudiare quanto registrato dalla Blockchain. Per fare questo lo strumento principale risulta essere lo smart contract che nello specifico caso di Hyperledger Fabric prende il nome di chaincode.

Il chaincode è un programma che può essere scritto in Go, node.js o Java e che implementa un'interfaccia prescritta. Un chaincode in genere gestisce la logica aziendale concordata dai membri della rete, secondo le normali logiche degli smart contract utilizzati nelle reti Blockchain. È possibile utilizzare un chaincode per aggiornare o interrogare il ledger di un canale. In Fabric il chaincode viene eseguito in un processo separato dal peer e inizializza e gestisce lo stato del registro attraverso le transazioni inviate dalle applicazioni.

Un prototipo di chaincode realizzato per questo specifico caso d'uso è mostrato nella figura sotto.

```
async refill(stub, args) {
  if (args.length != 7) {
  throw new Error('Incorrect number of  arguments'); }
  // identity of the smart contract submitter
  const creator = stub.getCreator();

  // certificate of smart contract submitter
  const cert = creator.id_bytes.toString('utf8');
  // name of smart contract submitter
  const mspid = creator.mspid.toString().split(
     /(?=[A-Z]let)/).map(s => s.toLowerCase());
  const mspid_name = mspid[0];
  // client name plus machine id
  const clientId = args[0];
  const clientIdSub = clientId.substring(0,7); // client name only

  const machineAsBytes = await stub.getState(args[0]);
  const machine = JSON.parse(machineAsBytes);
  // only "ClientX" can invoke the smart contract
  if(clientIdSub=='clientx'){
    // allowed: update values
    machine.barcode= args[1];
    machine.quantity= args[2];
    machine.scanTimestamp= args[3];
    machine.refillTimestamp= args[4];
    machine.producer= args[5];
    machine.contractId= args[6];

    // transaction generation
    await stub.putState(args[0], Buffer.from(JSON.stringify(machine)));
  }
  else{ console.info("STOP! not allowed."); }
}
```

In figura sono mostrati gli aspetti principali del prototipo di chaincode realizzato con il nome di Refill e scritto in node.js.

Inizialmente il codice verifica che siano stati inseriti tutti i parametri (in questo caso 7) e in caso contrario termina l'esecuzione rilanciando un errore. Successivamente viene controllato che il chaincode sia stato invocato da un mittente autorizzato. A tal fine, recupera l'identità del mittente del chaincode e verifica che il suo nome inizi con una determinata stringa, ad esempio "ClientX". In questo modo è possibile verificare che la creazione della transazione sia invocata

solamente da nodi appartenenti all'organizzazione autorizzata (tipicamente la gelateria) mentre i clienti di altre organizzazioni non possono richiedere la creazione di una nuova transazione per la specifica macchina da gelato. Se l'utente risulta autorizzato, il chaincode recupera le informazioni richieste, ovvero l'identificatore della ricarica tramite il codice a barre, la quantità di prodotto di ricarica utilizzato, i tempi di scansione del codice a barre e dell'operazione di refill, il nome del produttore della ricarica e l'identificativo del contratto stipulato. Infine, il chaincode genera la richiesta di transazione che verrà poi sottoposta ai nodi endorser.

4.3. Funzionalità applicazione Web

Per quanto riguarda la parte di progettazione dell'applicazione web si è pensato di realizzare un'applicazione (utilizzando il framework Ruby on Rails) con lo scopo principale di interagire direttamente con la Blockchain per consentire ad un cliente la visualizzazione dei refill effettuati dalle proprie macchine. L'applicazione web intende fornire le funzionalità necessarie per poter gestire l'accesso degli utenti attraverso il login, così da fornire una gerarchia di funzionalità sulla base dei permessi impostati.

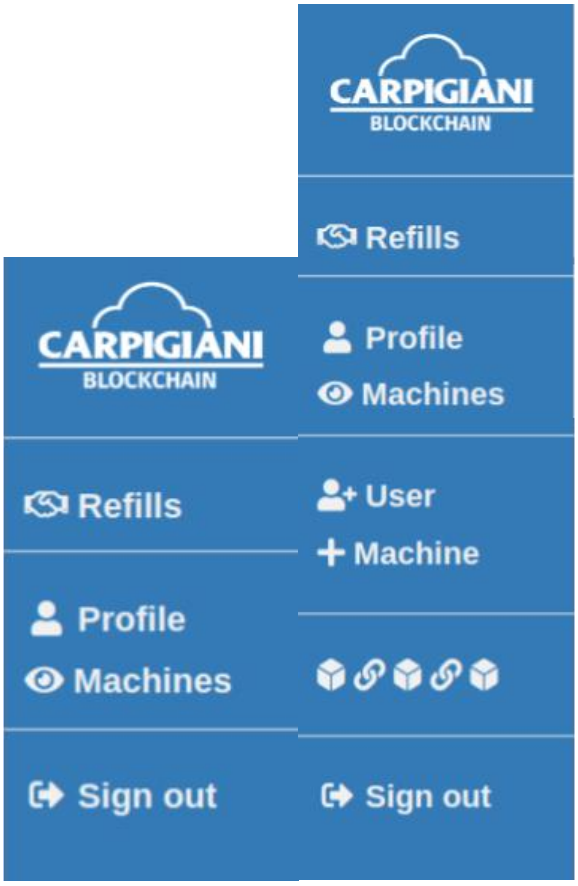
Si prevedono i seguenti profili utenti per l'utilizzo dell'applicazione Web:

- **utenti standard**: sono utenti abilitati a visionare i refill delle sole proprie macchine alle quali sono stati precedentemente associati da un utente amministratore;
- **amministratori**: sono particolari utenti che hanno la possibilità di aggiungere nuovi utenti (sia standard che amministratori), aggiungere nuove macchine (in associazione all'aggiunta delle organizzazioni nella rete blockchain proprietarie delle macchine inserite nell'applicazione) e aggiungere nuove relazioni tra un utente e una o più macchine, poiché, come detto, lo scopo primario dell'applicazione è quello di fornire accesso alle informazioni sui refill compiute dalle macchine al quale l'utente è associato. Si è previsto che l'organizzazione Carpigiani possa essere la sola organizzazione dotata di amministratori, ai quali è inoltre consentito di visionare i refill effettuati da qualsiasi macchina appartenente alla rete Blockchain (quindi di qualsiasi organizzazione) sia per un monitoraggio delle macchine che per un controllo su eventuali frodi.

È consentito l'accesso all'applicazione solo previo login con le proprie credenziali.

Di seguito viene presentato un Proof of Concept dell'interfaccia grafica che verrà implementata.

Nella figura sotto sono presentati i menù principali dell'interfaccia grafica. A sinistra è presentato il menu visualizzabile da un utente standard mentre a destra è mostrato il menu relativo ad un utente admin.



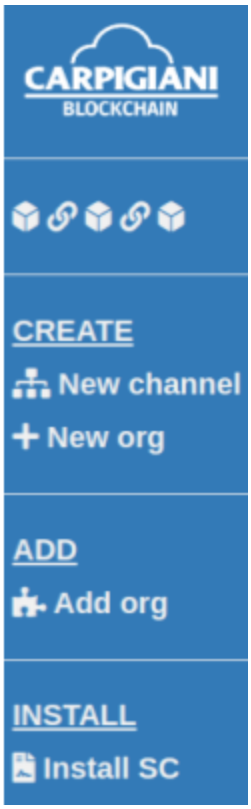
La funzionalità comuni a entrambe le tipologie di utenti sono:

- Refills: consente la visualizzazione dei refill afferenti alle proprie macchine (caso utenti standard) o relative a tutte le macchine (caso utenti admin).
- Profile: permette la modifica delle informazioni relative all'utente loggato
- Machines: mostra l'elenco delle macchine associate all'organizzazione dell'utente loggato (caso utente standard) o di tutte le macchine (nel caso di di utente amministratore)

In aggiunta alle funzionalità presentate sopra, l'utente amministratore è in grado di svolgere anche le seguenti funzionalità:

- aggiunta di un nuovo utente
- aggiunta di una nuova macchine
- entrare nella sezione riservata alla modifica della blockchain

A tal proposito nella figura sotto viene presentato anche il menu relativo alla sezione di modifica della blockchain.

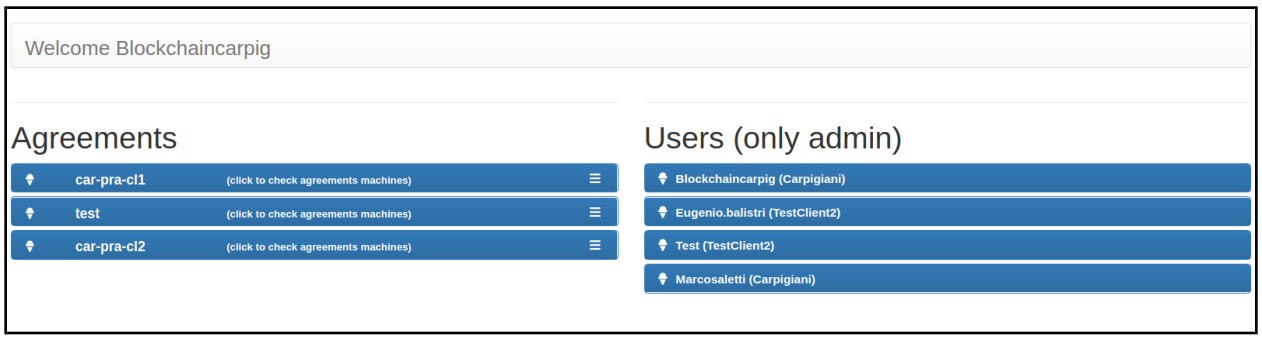


Le funzionalità rese disponibili dal menu di modifica della blockchain sono:

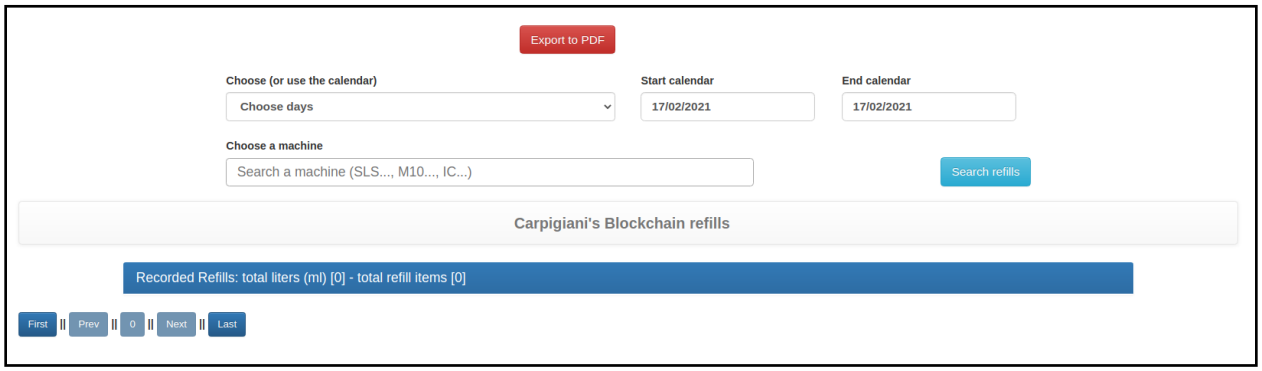
- la creazione di un nuovo canale;
- la creazione di una nuova organizzazione
- l'aggiunta di un'organizzazione ad un canale
- l'installazione di un nuovo Smart Contract (chaincode) su un canale.

Una volta effettuato il login un utente non amministratore visualizzerà i canali nei quali la propria organizzazione è registrata (nella sezione Agreements), come mostrato nella figura sotto.

Se invece il login viene effettuato da un utente amministratore, questi potrà visualizzare l'elenco di tutti i canali presenti nel sistema (nella sezione Agreements) e l'elenco degli utenti registrati (sezione Users), come evidenziato nella figura sotto.



Una volta selezionato il canale desiderato un utente potrà visionare i refill relativi ad una macchina impostando l'intervallo temporale della ricerca e il nome della macchina desiderata. Un esempio di questa schermata è mostrato nella figura sotto.



I dati relativi ad un refill, invece, sono mostrati nell'esempio della figura sotto, dove si mette in luce la presenza dei campi:

- Machine Refill: identifica la macchina che ha effetuato il refil;
- Contract ID: si riferisce allo smart contract utilizzato;
- Refill Quantity: indica la quantità di prodotto inserito nelle macchine;
- Refill Timestamp: è relativo all'istante nel quale avviene l'operazione di refill;
- Barcode: identifica il codice a barre presente nella scatola di miscela da gelato;
- Barcode Timestamp: si riferisce all'istante temporale nel quale viene letto il codice a barre;
- No. of cones: indica il numero di coni gelato realizzati dalla macchina fino a quel momento.

Infine, per quanto riguarda le funzionalità dell'amministratore, selezionando un canale sarà possibile vedere alcune informazioni relative alle organizzazioni che lo compongono, come ad esempio il nome del nodo Endpoint per ogni organizzazione, la lunghezza del Ledger o il nome dello Smart Contract installato, come mostrato nella figura sotto.