RdS
RICERCA DI SISTEMA

ENEA
AGENZIA NAZIONALE PER LE NUOVE TECNOLOGIE, L'ENERGIA E LO SVILUPPO ECONOMICO SOSTENIBILE

MINISTERO DELL'AMBIENTE E DELLA SICUREZZA ENERGETICA

Ricerca di Sistema elettrico

Consolidamenti di alcuni casi d'uso e sviluppo di ulteriori funzionalità per servizi di comunità energetiche (LA1.17)

M. Gallucci, M. Squillace

SoftStrategy
DRIVE THE VALUE

RdS_PTR 22-24_PR1.7_LA1.17_335

Consolidamenti di alcuni casi d'uso e sviluppo di ulteriori funzionalità per servizi di comunità energetiche

LA 1.17 - Servizi social per le CER: L'interfaccia web del Marketplace per la LTE e l'integrazione di ECListener

M. Gallucci, M. Squillace (Soft Strategy)

Dicembre 2024

Report Ricerca di Sistema Elettrico

Accordo di Programma Ministero dell'Ambiente e della Sicurezza Energetica - ENEA Piano Triennale di Realizzazione 2022-2024

Obiettivo: Decarbonizzazione

Progetto: Tema di ricerca 1.7 – Tecnologie per la penetrazione efficiente del vettore elettrico negli usi finali

Linea di attività: 1.17

Responsabile del Progetto: Claudia Meloni, ENEA

Responsabile del Work Package: Angelo Frascella, ENEA

Responsabile Linea di Attività: Gilda Massa (ENEA)

Mese inizio previsto: 19

Mese inizio effettivo: 19

Mese fine previsto: 36

Mese fine effettivo: 36

Il presente documento descrive le attività di ricerca svolte all'interno contratto dal titolo: Consolidamenti di alcuni casi d'uso e sviluppo di ulteriori funzionalità per servizi di comunità energetiche

Indice

1 Risultati attesi 5
1.1 Miglioramento dell'accessibilità e sicurezza 5
1.2 Ottimizzazione delle prestazioni della blockchain 5
1.3 Miglioramento dell'esperienza utente 5
1.4 Maggiore flessibilità nel marketplace 5
1.5 Implementazione di un workflow di approvazione per i prodotti 5
1.6 Risoluzione di bug 5
2 Risultati ottenuti 6
2.1 Integrazione con il nuovo Identity Provider di Enea 6
2.2 Ottimizzazione delle prestazioni della blockchain 7
2.3 Miglioramento dell'esperienza utente 7
2.4 Maggiore flessibilità nel marketplace 8
2.5 Implementazione di un workflow di approvazione per i prodotti 8
2.6 Risoluzione di bug 8
3 Prodotti attesi 9
4 Prodotti sviluppati 10
4.1.1 Platform Admin 10
4.1.1.1 Homepage 10
4.1.1.2 Registrazione al servizio 11
4.1.1.3 Aggiungi comunità 12
4.1.2 Community Member 13
4.1.2.1 Marketplace 13
4.1.2.1.1 I beni e servizi disponibili 13
4.1.2.1.2 Riepilogo acquisto bene 14
Analisi degli scostamenti su attività e risultati 15
5 Sintesi delle attività svolte 16
6 Dettaglio delle attività svolte 17
6.1 Ambiente, framework e librerie 19
6.1.1 DSM (BE) 19
6.1.2 BLOCKCHAIN 20
6.1.3 DAPP (FE) 20

6.2 DSM (BE) 2
6.2.1 Elenco delle API 2
6.2.2 Descrizione delle risorse statiche 2
6.2.3 DMDB - Data Model 2
6.3 BLOCKCHAIN 4
6.3.1 Componenti: 4
6.3.2 Sviluppo 4
6.3.2.1 Migrazione al framework Hardhat 5
6.3.3 Interfacce e Contratti 6
6.3.4 Struttura del Codice 7
6.3.5 Fase di Test 7
6.3.6 Fase di Deploy 7
7 Contributo delle eventuali consulenze alle attività sopra descritte 8
8 Pubblicazioni scientifiche 9
9 Eventi di disseminazione 10

Indice delle figure

Figura 1 – Schema architetturale 6
Figura 2 – Schema logico IdP 6
Figura 3 - Platform Admin – Homepage in fase di sviluppo 10
Figura 5 - Registrazione 11
Figura 6 - Aggiungi comunità 12
Figura 12 - Marketplace (utilizzo) 13
Figura 13 - Marketplace - Riepilogo acquisto bene 14
Figura 19 - Struttura delle risorse statiche del progetto 2
Figura 20 - Modello Logico del DB 3
Figura 21 - Framework Hardhat per Deplyment e Testing degli Smart Cotnract 5
Figura 22 - Schema di deployment e interazione con lo Smart Contract 6

Indice delle tabelle

Tabella 1 – Callback IdP 7
Tabella 2 – DSM (BE)– Ambiente 19
Tabella 3 – DSM (BE) - Framework e librerie 19
Tabella 4 – Blockchain - Ambiente 20
Tabella 5 – Blockchain – Framework e librerie 20
Tabella 6 – DAPP (FE) - Ambiente 20
Tabella 7 – DAPP (FE) – Framework e librerie 20
Tabella 8 – API Marketplace 4
Tabella 9 – API Associazione Modelli 5
Tabella 10 – API Gestione Modelli 6
Tabella 11 – API Gestione Strutture 7
Tabella 12 – API KPI della Piattaforma 8
Tabella 13 – API KPI della Comunità 10

1 Risultati attesi

Nei paragrafi seguenti viene riportata una lista dei risultati attesi come da capitolato vigente.

1.1 Miglioramento dell'accessibilità e sicurezza

È stata richiesta l'integrazione con il nuovo Identity Provider (IDP) di Enea per una gestione più sicura ed efficiente degli accessi, rafforzando i meccanismi di autenticazione e migliorando l'affidabilità della piattaforma.

1.2 Ottimizzazione delle prestazioni della blockchain

È stata richiesta un'ottimizzazione delle prestazioni della blockchain con una soluzione più performante che riduca i tempi di risposta e migliori l'efficienza complessiva della piattaforma, consentendo una gestione più rapida delle transazioni.

1.3 Miglioramento dell'esperienza utente

È stato richiesto l'adeguamento agli standard AGID per assicurare la conformità alle normative italiane, favorendo un accesso inclusivo per tutti gli utenti. Inoltre, è stato richiesto il re-factoring delle principali pagine di riepilogo delle informazioni al fine di ottenere un livello di dettaglio maggiore e una visualizzazione più intuitiva del wallet e delle risorse disponibili.

1.4 Maggiore flessibilità nel marketplace

Sono state richieste nuove funzionalità che permettano agli utenti di modificare i prezzi dei beni e servizi già inseriti nel marketplace, rendendo l'offerta più dinamica, e di allegare immagini ai prodotti inseriti, arricchendo la presentazione visiva e fornendo informazioni più complete.

Si richiede, inoltre. di abilitare i soggetti detentori di strutture alla prenotazione e/o acquisto di beni e servizi nel marketplace. Infine, sono stati richiesti filtri avanzati per facilitare la ricerca di beni e servizi all'interno del marketplace, migliorando la fruizione della piattaforma.

1.5 Implementazione di un workflow di approvazione per i prodotti

È stata richiesta l'introduzione di un sistema di approvazione per i prodotti pubblicati per garantire un maggiore controllo sui prodotti pubblicati nel marketplace. Il community-admin riceverà una notifica via e-mail per ogni nuovo prodotto da approvare, con la possibilità di visionarne i dettagli e accettarlo o rifiutarlo. Parimenti, il community-member riceverà una notifica via e-mail dell'avvenuta pubblicazione del prodotto da egli inserito.

1.6 Risoluzione di bug

È stata richiesta la risoluzione di alcuni bug che impediscono l'utilizzo ottimale della piattaforma.

2 Risultati ottenuti

La soluzione proposta ha ampliato e migliorato l'attuale piattaforma software, introducendo nuove funzionalità volte a ottimizzare il funzionamento della comunità energetica. Le modifiche implementate mirano a potenziare l'efficienza dei servizi, migliorare la sicurezza e garantire una gestione più intuitiva e accessibile.

Di seguito viene illustrato il nuovo schema logico architetturale e i risultati raggiunti.

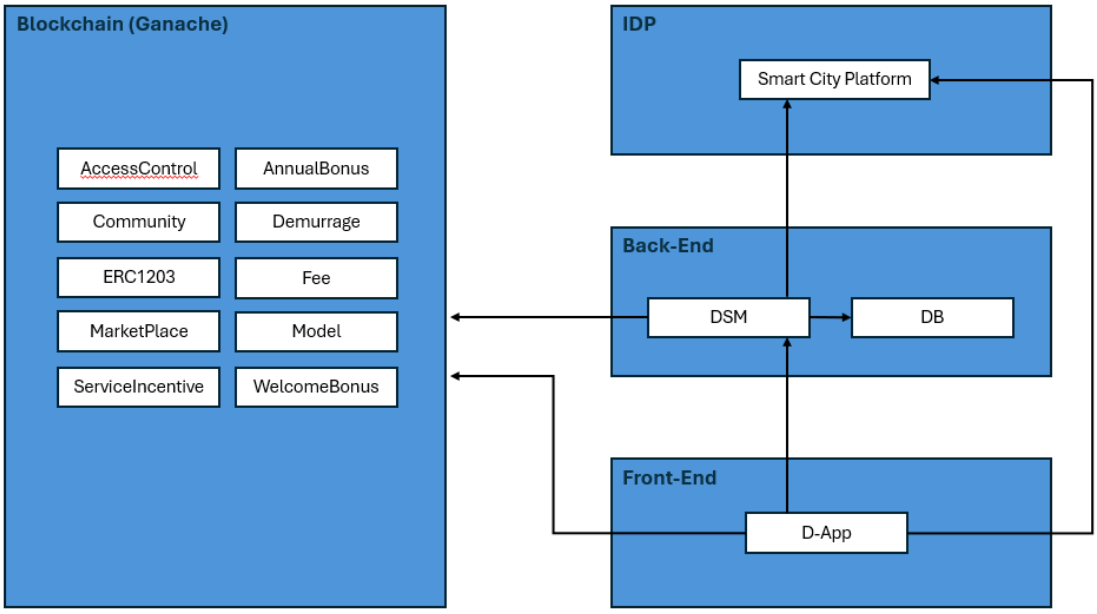

Figura 1 – Schema architetturale

2.1 Integrazione con il nuovo Identity Provider di Enea

Le componenti back-end e front-end sono state aggiornate per gestire l'integrazione con il nuovo Identity Provider (Smart City Platform) fornito da ENEA, che sostituisce il precedente Identity Provider (Winter Bologna). Questa integrazione ha migliorato la gestione delle identità digitali e la sicurezza dell'accesso alla piattaforma.

L'implementazione ha previsto che, per le operazioni di registrazione e login, l'utente venga rediretto dal front-end all'identity provider, che gestisce tali operazioni in modo centralizzato. Una volta completato il processo di registrazione e/o autenticazione, l'identity provider invoca una callback esposta dal front-end per reindirizzare l'utente all'interno della piattaforma, garantendo un'esperienza fluida e sicura.

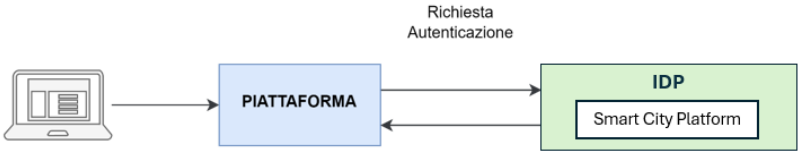

Figura 2 – Schema logico IdP

Lista di url callback esposte dal front-end:

Nome	Url	Descrizione
registerSuccess	/registerSuccess	Questa URL di callback esposta dal front-end è necessaria a far tornare l'utente nella piattaforma dopo che esso ha effettuato la registrazione all'interno della piattaforma dell'identity provider.
loginSuccess	/loginSuccess?token={jwt}	Questa URL di callback esposta dal front-end è necessaria a far tornare l'utente nella piattaforma dopo che esso ha effettuato il login all'interno della piattaforma dell'identity provider.

Tabella 1 – Callback IdP

2.2 Ottimizzazione delle prestazioni della blockchain

L'ottimizzazione delle prestazioni della blockchain ha portato a significativi miglioramenti in termini di efficienza, tempi di risposta e gestione delle transazioni. Una delle principali azioni implementate è stata la migrazione verso una blockchain privata basata su Ganache, istanziata all'interno di un container Docker. Questa scelta tecnologica ha consentito di ottenere un ambiente più controllato, stabile e performante, garantendo una maggiore reattività del sistema e una gestione ottimizzata delle risorse.

Grazie a questa migrazione, i tempi di elaborazione delle transazioni sono stati drasticamente ridotti, migliorando l'esperienza utente e consentendo operazioni più rapide e fluide. L'adozione di Ganache ha permesso di simulare un ambiente blockchain con maggiore efficienza rispetto alla soluzione precedente, eliminando le latenze. Inoltre, l'uso di un container Docker ha semplificato la gestione e la scalabilità dell'infrastruttura, facilitando il deployment e la manutenzione del sistema.

2.3 Miglioramento dell'esperienza utente

Al fine di migliorare complessivamente l'esperienza utente, sono state introdotte una serie di migliorie in diverse sezioni dell'applicativo.

In particolare:

Adeguamento alla normativa AGID: Per garantire la piena conformità della piattaforma alle linee guida AGID (Agenzia per l'Italia Digitale), sono stati introdotti miglioramenti strutturali e funzionali. L'aggiornamento ha incluso l'adeguamento dell'interfaccia ai criteri di accessibilità WCAG 2.1, garantendo una migliore esperienza per gli utenti con disabilità.

Restyling della pagina del profilo utente: La pagina profilo è stata ridisegnata per offrire una visione più chiara e organizzata delle informazioni personali, con sezioni distinte per dati anagrafici e wallet.

Re-factoring delle pagine di riepilogo delle transazioni: Miglioramento della fruibilità: Il layout delle pagine di riepilogo delle transazioni è stato riprogettato per una consultazione più intuitiva. Le informazioni sono ora strutturate in modo chiaro, riducendo la necessità di navigazioni aggiuntive.

Sezione wallet ottimizzata: È stata migliorata la visualizzazione delle risorse possedute dagli utenti, con dettagli aggiuntivi sulle transazioni recenti. Inoltre, è stato introdotto un riepilogo delle operazioni categorizzato per tipologia di transazione, facilitando la gestione finanziaria.

2.4 Maggiore flessibilità nel marketplace

Al fine di estendere le possibilità di utilizzo del marketplace, sono state aggiunte nuove funzionalità, tra le quali:

Maggiore flessibilità per gli utenti: È stata introdotta una funzionalità che consente agli utenti di modificare il prezzo di un bene o servizio già inserito nel marketplace, permettendo un aggiornamento dinamico delle offerte.

Interfaccia intuitiva: L'utente può accedere alla sezione di modifica direttamente dalla pagina di gestione degli annunci. Il sistema guida l'utente nella modifica attraverso campi precompilati e opzioni di conferma per evitare errori.

Accesso esteso alle funzionalità del marketplace: i soggetti detentori di strutture possono ora prenotare e acquistare beni e servizi direttamente dalla piattaforma, usufruendo di tutte le funzionalità disponibili per i community-member.

Miglioramento della ricerca: Sono stati introdotti filtri di ricerca avanzati all'interno del marketplace che consentono agli utenti di trovare beni e servizi in modo più rapido ed efficace. La ricerca può essere affinata per categoria e tipologia, migliorando la navigazione.

Supporto alle immagini nei prodotti: Gli utenti ora possono allegare foto ai propri annunci, aumentando l'attrattiva visiva e migliorando la qualità delle inserzioni. Questo aggiornamento facilita le decisioni d'acquisto e arricchisce l'esperienza degli utenti.

2.5 Implementazione di un workflow di approvazione per i prodotti

Ogni prodotto pubblicato nel marketplace passa, ora, attraverso un flusso di approvazione gestito dal back-end. Il community-admin riceve automaticamente una notifica via e-mail ogni volta che un nuovo prodotto è in attesa di revisione. Il community-admin può visualizzare tutti i dettagli del prodotto (descrizione, immagini, prezzo, ecc.) prima di approvarlo o respingerlo.

Dopo la decisione del community-admin, il proprietario del prodotto riceve un'e-mail con l'esito della revisione, assicurando trasparenza e comunicazione efficace.

2.6 Risoluzione di bug

Sono state apportate migliorie generali all'applicativo, risolvendo alcuni bug che impedivano il corretto utilizzo della piattaforma. Questi interventi contribuiranno a rendere la piattaforma più stabile, sicura e performante.

3 Prodotti attesi

Nell'ambito di questa evolutiva, sono attesi aggiornamenti sui seguenti prodotti del progetto:

1. **ENEA-DAPP (Front-end)**
   L'applicazione web dovrà essere aggiornata per migliorare l'esperienza utente e supportare le nuove funzionalità richieste. Gli interventi previsti includono:
   - **Integrazione con il nuovo Identity Provider**, garantendo un flusso di autenticazione più sicuro e centralizzato.
   - **Restyling della UI**, con adeguamento alle linee guida AGID e miglioramenti dell'accessibilità.
   - **Ottimizzazione della sezione wallet**, con una gestione più chiara delle transazioni e delle risorse possedute.
   - **Miglioramento della gestione del marketplace**, con l'introduzione del supporto alle immagini, la possibilità di modificare i prezzi e nuovi filtri di ricerca.
   - **Implementazione di un workflow di approvazione**, per permettere ai community-admin di validare gli annunci prima della pubblicazione.

2. **DSM (Back-end)**
   Il back-end dovrà essere adeguato per supportare le nuove funzionalità e garantire maggiore stabilità e sicurezza. Le modifiche previste comprendono:
   - **Implementazione dell'integrazione con il nuovo Identity Provider**, con gestione ottimizzata delle sessioni e delle autorizzazioni.
   - **Estensione delle API del marketplace**, per abilitare la modifica dinamica dei prezzi, il processo di prenotazione e l'approvazione degli annunci.
   - **Miglioramenti nella gestione delle prenotazioni**, introducendo vincoli su date e orari per evitare selezioni errate.
   - **Ottimizzazione delle performance**, per ridurre i tempi di risposta e migliorare la scalabilità del sistema.

- **ENEA-CONTRACT (Blockchain)**

  Per garantire maggiore efficienza e stabilità, saranno apportati aggiornamenti alla componente blockchain. Gli interventi previsti includono:
  - **Migrazione a una blockchain privata basata su Ganache**, migliorando il controllo e le prestazioni delle transazioni.
  - **Adeguamento degli smart contract**, per supportare le nuove logiche di pricing e prenotazione nel marketplace.
  - **Ottimizzazione del tracciamento delle transazioni**, per garantire maggiore trasparenza e auditabilità.

4 Prodotti sviluppati

Nei paragrafi seguenti vengono mostrate alcune sezioni della piattaforma che sono state oggetto del restyling al fine di essere conformi alle adeguative AGID. L'obiettivo è di illustrare l'aspetto e il flusso delle funzionalità progettate per l'applicazione. I mockup offrono una visione preliminare della struttura e dell'organizzazione dei contenuti, consentendo di valutare l'esperienza utente e di garantire coerenza visiva e funzionale rispetto ai requisiti del progetto come richiesto nelle specifiche ENEA.

4.1 Platform Admin

4.1.1 Homepage

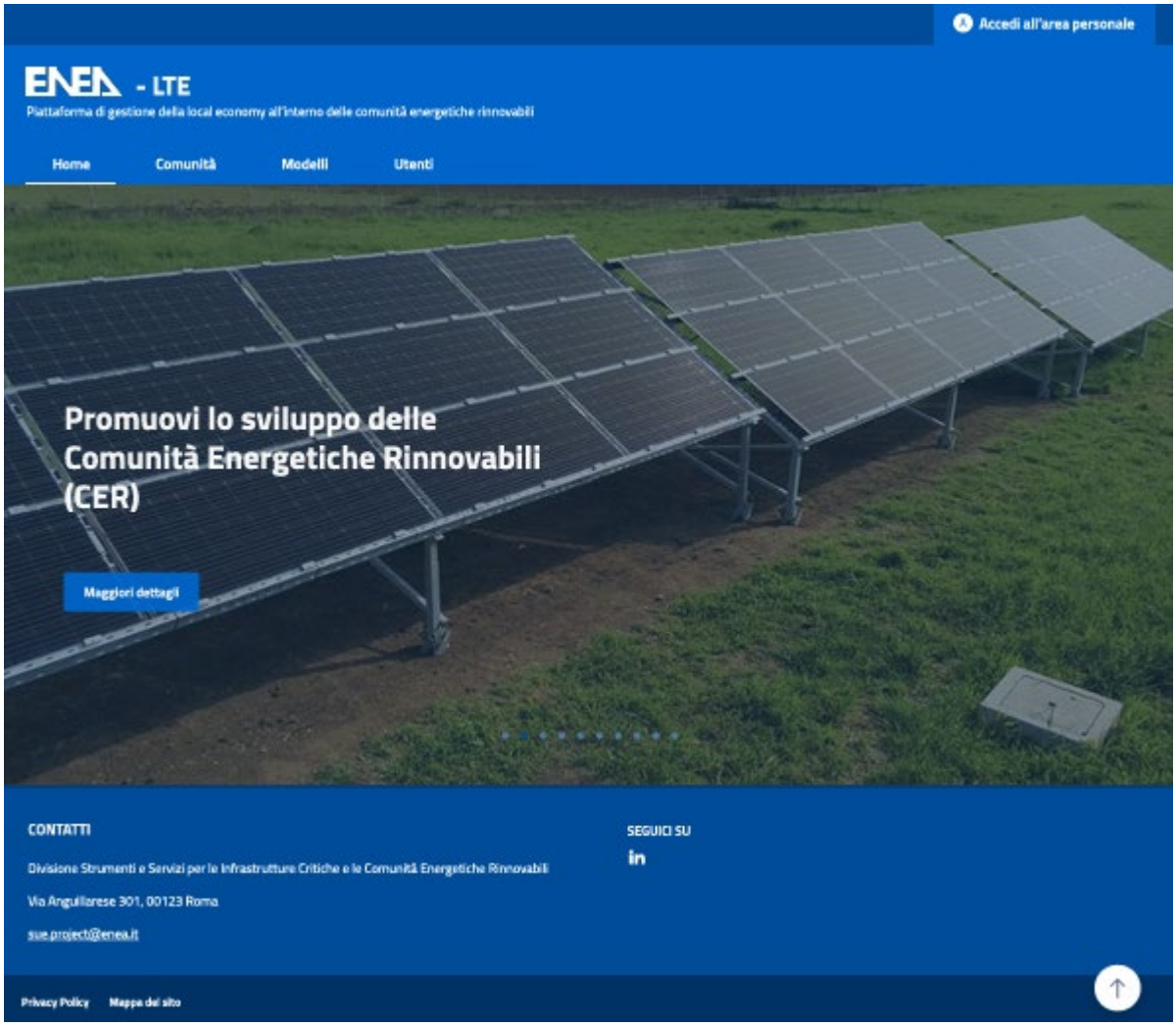

Figura 3 - Platform Admin – Homepage in fase di sviluppo

4.1.2 Registrazione al servizio

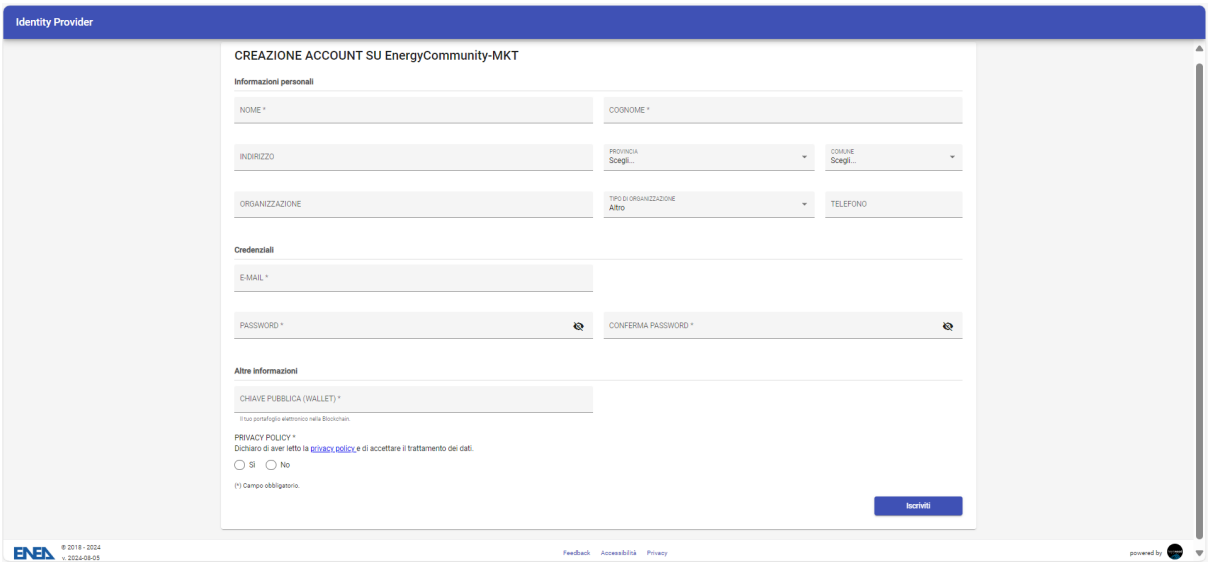

Figura 4 - Registrazione

4.1.3 Aggiungi comunità

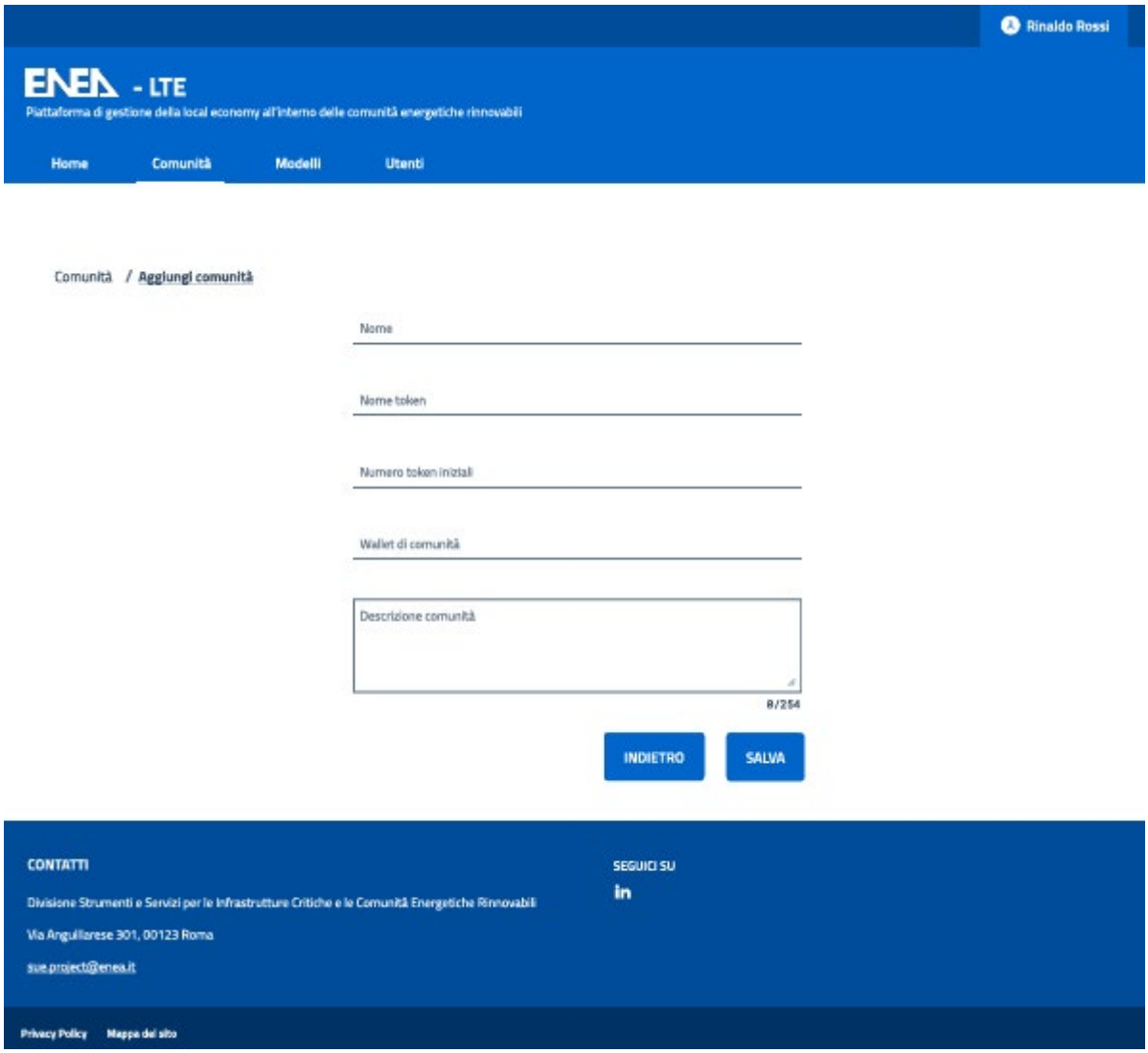

Figura 5 - Aggiungi comunità

4.1.4 Community Member

4.1.4.1 Marketplace

4.1.4.1.1 I beni e servizi disponibili

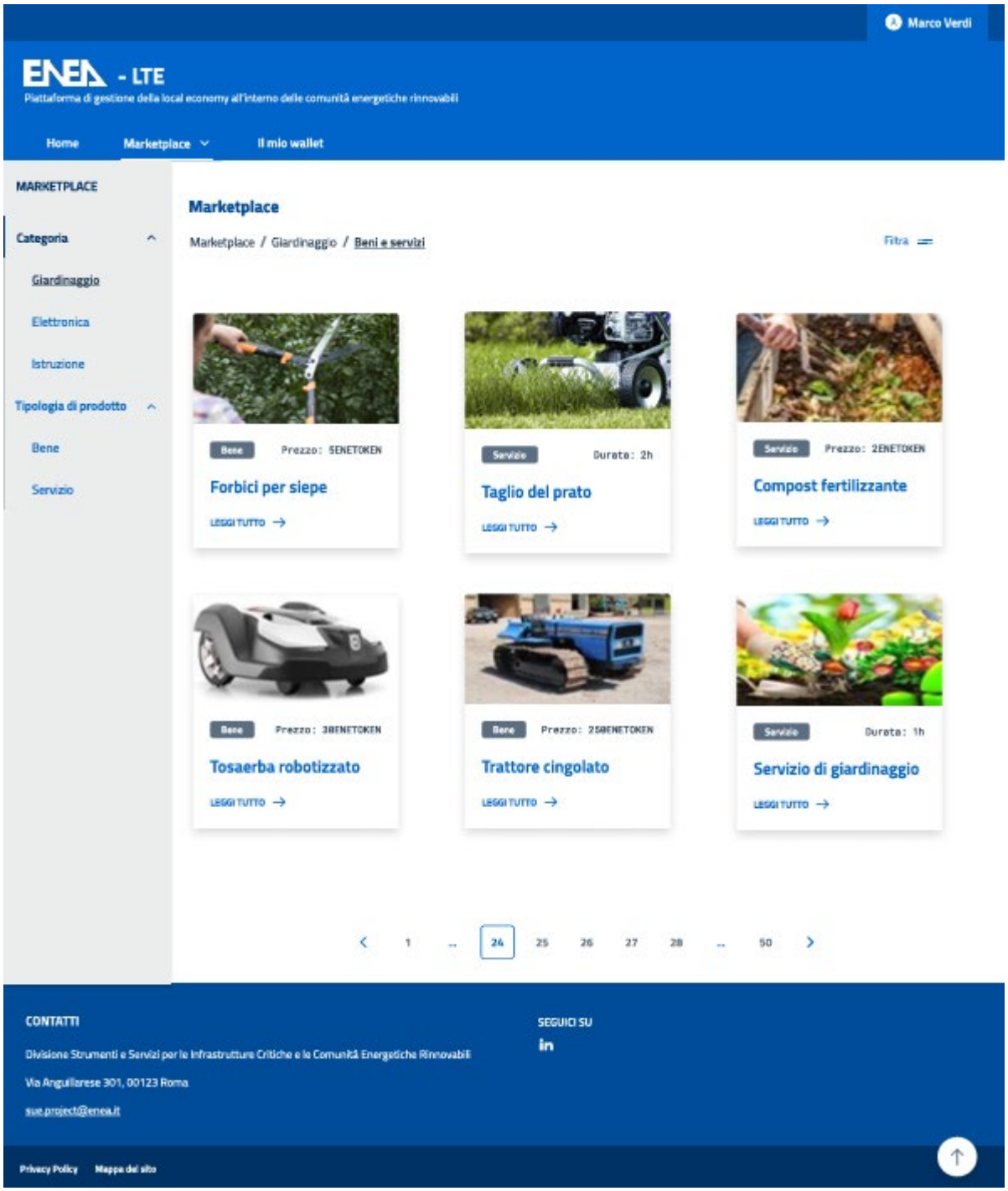

Figura 6 - Marketplace (utilizzo)

4.1.4.1.2 Riepilogo acquisto bene

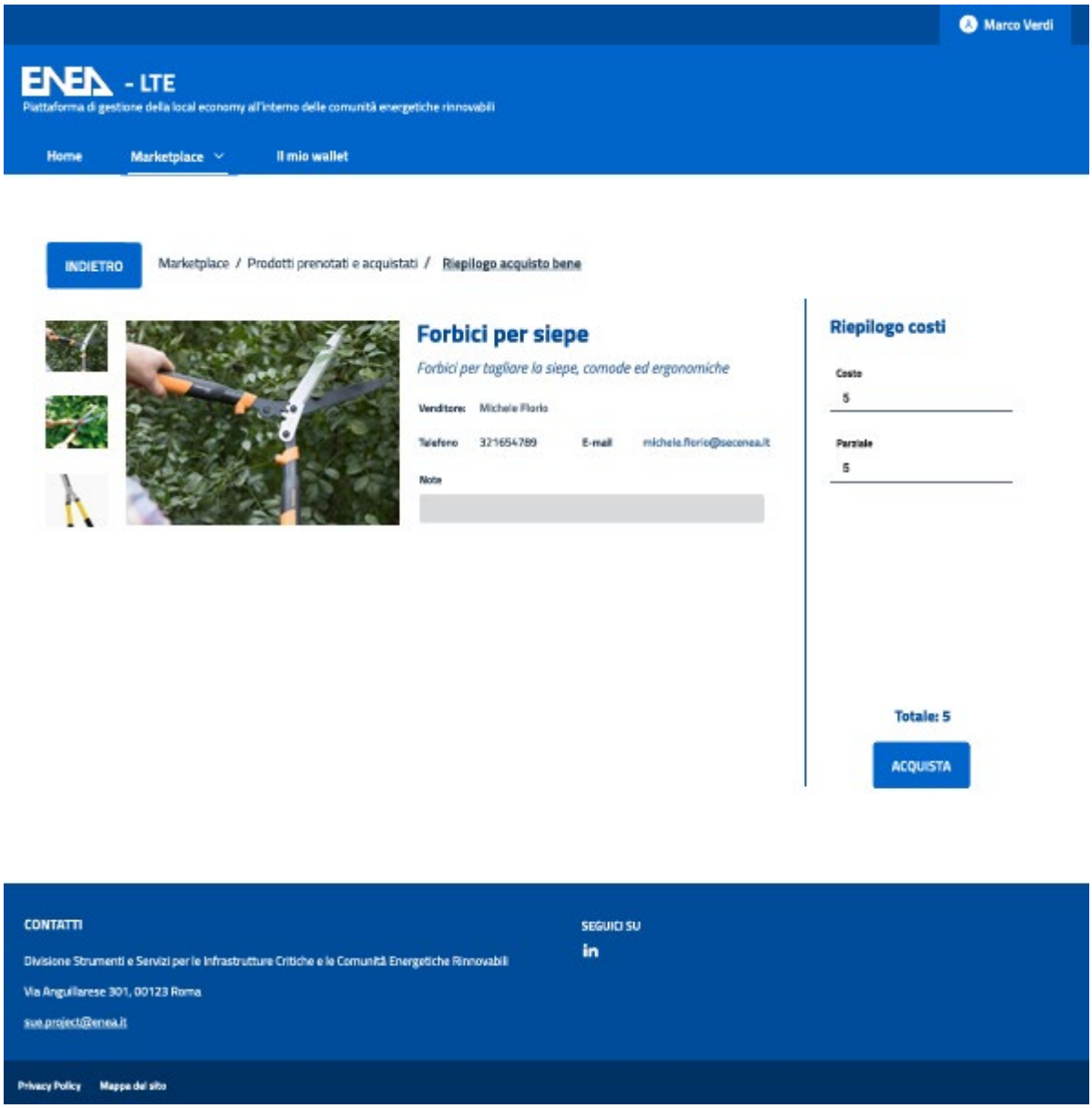

Figura 7 - Marketplace - Riepilogo acquisto bene

Analisi degli scostamenti su attività e risultati

Non sono stati riscontrati scostamenti tecnici/economici rispetto al preventivo.

5 Sintesi delle attività svolte

Si riporta di seguito la sintesi delle attività svolte e dei risultati ottenuti in relazione ai risultati attesi:

La piattaforma software della comunità energetica è stata migliorata con nuove funzionalità per ottimizzare servizi, sicurezza e usabilità.

Integrazione con il nuovo Identity Provider: È stato aggiornato il sistema di autenticazione per utilizzare l'Identity Provider di ENEA, garantendo maggiore sicurezza e una gestione centralizzata delle identità digitali.

Ottimizzazione della blockchain: La piattaforma è stata migrata a una blockchain privata basata su Ganache, eseguita in un container Docker. Questo ha ridotto i tempi di elaborazione delle transazioni, migliorando stabilità e scalabilità.

Miglioramento dell'esperienza utente: Sono stati apportati aggiornamenti per conformità AGID, accessibilità WCAG 2.1, restyling della pagina profilo, riorganizzazione delle transazioni e ottimizzazione della sezione wallet.

Marketplace più flessibile: È stata introdotta la modifica dinamica dei prezzi, un'interfaccia intuitiva per la gestione degli annunci, accesso ampliato alle funzionalità di prenotazione e acquisto, filtri di ricerca avanzati e supporto per immagini nei prodotti.

Workflow di approvazione per i prodotti: Ora ogni prodotto nel marketplace deve essere approvato dal community-admin, con notifiche automatiche.

Risoluzione di bug minori: Sono stati corretti problemi che compromettevano il corretto utilizzo di alcune interfacce della piattaforma.

Queste migliorie hanno reso la piattaforma più efficiente, sicura e intuitiva per gli utenti.

6 Dettaglio delle attività svolte

Di seguito è riportato il dettaglio delle attività svolte:

Integrazione con il nuovo Identity Provider

L'infrastruttura di autenticazione è stata aggiornata per supportare il nuovo Identity Provider di ENEA (Smart City Platform), sostituendo il precedente (Winter Bologna). Il sistema di login e registrazione è stato modificato affinché gli utenti vengano reindirizzati all'Identity Provider, che gestisce l'autenticazione in modo centralizzato. Una volta completata, l'utente viene riportato sulla piattaforma tramite una callback dedicata. Questo aggiornamento ha migliorato la sicurezza, ridotto la complessità della gestione delle identità e reso l'esperienza di accesso più fluida.

Ottimizzazione della blockchain

Per migliorare l'efficienza del sistema, la piattaforma è stata migrata a una blockchain privata basata su Ganache, eseguita in un container Docker.

Questo ha permesso:

Un ambiente più stabile e controllato.
Una riduzione significativa dei tempi di elaborazione delle transazioni.
La minimizzazione delle latenze rispetto alla soluzione precedente.
Una gestione delle risorse più efficiente e scalabile.
L'uso di Docker ha semplificato il deployment e la manutenzione, facilitando l'aggiornamento della piattaforma e l'adattamento a nuove esigenze.

Miglioramento dell'esperienza utente

Per rendere la piattaforma più intuitiva e conforme agli standard normativi, sono stati apportati i seguenti aggiornamenti:

Adeguamento AGID: L'interfaccia è stata adattata alle linee guida di accessibilità WCAG 2.1, garantendo una migliore fruibilità per utenti con disabilità.
Restyling della pagina profilo: Ora i dati anagrafici e il wallet sono organizzati in sezioni distinte per una consultazione più chiara.
Miglioramento delle pagine di riepilogo delle transazioni: Il layout è stato rivisto per una consultazione più rapida e intuitiva, riducendo il numero di passaggi necessari per accedere alle informazioni.
Ottimizzazione della sezione wallet: Sono stati aggiunti dettagli sulle transazioni recenti e una categorizzazione per tipologia, facilitando il monitoraggio delle risorse economiche.

Maggiore flessibilità nel marketplace

Sono state introdotte nuove funzionalità per migliorare l'interazione con il marketplace:

Modifica dinamica del prezzo: Gli utenti possono aggiornare il prezzo dei beni e servizi inseriti, rendendo le offerte più flessibili.

Interfaccia di gestione migliorata: L'area di modifica degli annunci è stata resa più intuitiva, con campi precompilati e opzioni di conferma per ridurre gli errori.

Accesso esteso alle funzionalità di prenotazione e acquisto: i soggetti detentori di strutture possono ora interagire direttamente con il marketplace, prenotando o acquistando prodotti e servizi.

Filtri di ricerca avanzati: Gli utenti possono cercare prodotti in base a categorie e tipologie, migliorando la navigabilità.

Supporto per immagini nei prodotti: Ora gli annunci possono includere foto, aumentando l'attrattiva visiva e migliorando l'esperienza di acquisto.

Workflow di approvazione per i prodotti

È stato introdotto un meccanismo di approvazione per i prodotti inseriti nel marketplace:

Ogni nuovo annuncio deve essere esaminato dal community-admin prima di essere pubblicato.

Il community-admin riceve notifiche via e-mail e può visualizzare i dettagli del prodotto prima di approvarlo o respingerlo.

L'esito della revisione viene comunicato via e-mail all'utente che ha inserito il prodotto. Questo sistema garantisce maggiore controllo sulla qualità e affidabilità degli annunci.

Risoluzione di bug minori

Sono stati identificati e corretti vari bug che compromettevano alcune funzionalità della piattaforma, ad esempio, sono stati migliorati aspetti legati alla stabilità generale dell'applicazione, sono state apportate ottimizzazioni minori hanno reso la piattaforma più sicura e performante.

Grazie a queste attività, la piattaforma offre ora un'esperienza più affidabile, efficiente e sicura per tutti gli utenti.

Di seguito sono descritti i framework, le librerie e gli ambienti utilizzati per lo sviluppo delle singole componenti della piattaforma. Per ogni componente viene anche indicata la struttura del codice con la descrizione dei principali package e delle risorse statiche.

Tutto il codice consegnato come elaborato del progetto è corredato di documentazione dettagliata come da standard attuali.

6.1 Ambiente, framework e librerie

Nel seguente paragrafo vengono descritte tutte le specifiche tecniche degli ambienti, framework, librerie e altre informazioni delle funzionalità aggiuntive sviluppate.

6.1.1 DSM (BE)

Ambiente	Versione	Dependency Management
Java	11	Maven

Tabella 2 – DSM (BE)– Ambiente

Framework	Spring
Boilerplate	Spring Boot 2.4.4
Layer autorizzativo	Spring Security 5.4.5
Connettore Blockchain	Web3j 4.9.8
Driver Database	org.postgresql 42.2.5
Gestione Token su standard JWT	io.jsonwebtoken.jjwt 0.12.6
Gestione testi/Stringhe	org.apache.commons 1.9
Unit Test	Junit
Algoritmo di derivazione delle chiavi crittografiche	Custom
Riconoscimento formati file	org.apache.tika 2.9.2
Framework	Spring
Boilerplate	Spring Boot 2.4.4

Tabella 3 – DSM (BE) - Framework e librerie

6.1.2 BLOCKCHAIN

Ambiente	Versione	Dependency Management
Node	20.17.0	Npm

Tabella 4 – Blockchain - Ambiente

Framework	Hardhat 2.3.3
Libreria	@openzeppelin/contracts: ^4.0.0
Libreria	dotenv: ^16.4.5

Tabella 5 – Blockchain – Framework e librerie

6.1.3 DAPP (FE)

Ambiente	Versione	Dependency Management
Node	20.17.0	Npm

Tabella 6 – DAPP (FE) - Ambiente

Framework	Angular 18.2.0
Libreria	@angular/animations: ^18.2.0
Libreria	@angular/cdk: ^18.2.10
Libreria	@angular/common: ^18.2.0
Libreria	@angular/compiler: ^18.2.0,
Libreria	@angular/core: ^18.2.0,
Libreria	@angular/forms: ^18.2.0,
Libreria	@angular/platform-browser: ^18.2.0,
Libreria	@angular/platform-browser-dynamic: ^18.2.0,
Libreria	@angular/router: ^18.2.0,
Libreria	bootstrap: ^5.3.3,
Libreria	bootstrap-italia: ^2.9.1,
Libreria	design-angular-kit: ^1.1.0,

Tabella 7 – DAPP (FE) – Framework e librerie

6.2 DSM (BE)

6.2.1 Elenco delle API

Nel seguente paragrafo vengono descritte le API REST esposte dal back-end, sono evidenziate in verde quelle che sono state realizzate nell'oggetto dell'evolutiva 2024.

Route	Method	Description
/api/dsm/v1/community/{_communityId}/marketplace/serviceByUser	GET	Restituzione di tutti i prodotti prenotati da un utente
/api/dsm/v1/community/{_communityId}/marketplace/serviceCategory	GET	Restituzione della lista delle categorie
/api/dsm/v1/community/{_communityId}/marketplace/serviceCategory	POST	Creazione di una categoria
/api/dsm/v1/community/{_communityId}/marketplace/serviceCategory	POST	Modifica della categoria
/api/dsm/v1/community/{_communityId}/marketplace/serviceCategory/{_categoryId}	POST	Aggiunta/Modifica della categoria
/api/dsm/v1/community/{_communityId}/marketplace/serviceCategory/{_categoryId}	GET	Restituisce i dettagli della categoria

/api/dsm/v1/community/{_communityId}/marketplace/serviceCategory/{_categoryId}	DELETE	Cancellazione della categoria
/api/dsm/v1/community/{_communityId}/marketplace/serviceCategory/{_categoryId}/service	POST	Aggiunta di un prodotto all'interno della categoria
/api/dsm/v1/community/{_communityId}/marketplace/serviceCategory/{_categoryId}/service/{_serviceId}	POST	Modifica di un prodotto all'interno della categoria
/api/dsm/v1/community/{_communityId}/marketplace/serviceCategory/{_categoryId}/service/{_serviceId}	GET	Restituisce i dettagli del prodotto
/api/dsm/v1/community/{_communityId}/marketplace/serviceCategory/{_categoryId}/service/{_serviceId}	DELETE	Cancellazione di un prodotto dalla categoria
/api/dsm/v1/community/{_communityId}/marketplace/serviceCategory/{_categoryId}/service/{_serviceId}/book	PUT	Prenotazione di un prodotto
/api/dsm/v1/community/{_communityId}/marketplace/serviceCategory/{_categoryId}/service/{_serviceId}/pay	POST	Pagamento di un prodotto
/api/dsm/v1/community/{_communityId}/marketplace/serviceCategory/{_categoryId}/service/{serviceId}/approve	POST	Approvazione servizio
/api/dsm/v1/community/{_communityId}/marketplace/serviceCategory/{_categoryId}/service/{serviceId}/disapprove	POST	Disapprovazione servizio
/api/dsm/v1/community/{_communityId}/marketplace/serviceCategory/{categoryId}/service/{serviceId}	POST	Aggiornamento servizio
/api/dsm/v1/community/{_communityId}/marketplace/serviceCategory/{categoryId}/service/{serviceId}/image/{progressive}	GET	Recupera una delle immagini del servizio

/api/dsm/v1/community/{_communityId}/marketplace/services		Lista dei prodotti appartenenti ad una categoria
/api/dsm/v1/community/{_communityId}/marketplace/services/toApprove	GET	Lista deo prodotti da approvare
/api/dsm/v1/community/{_communityId}/marketplace/transactions	GET	Restituzione della lista di transazioni riguardanti una determinata comunità
/api/dsm/v1/community/{_communityId}	GET	Recupero della comunità
/api/dsm/v1/community/{_communityId}	PATCH	Aggiorna la comunità
/api/dsm/v1/community/{_communityId}	DELETE	Cancella la comunità
/api/dsm/v1/community/{_communityId}/user/{_userId}/year/{_year}/month/{_month}/day/{_day}	GET	Takeout
/api/dsm/v1/community/{_communityId}/wallet	PUT	Aggiornamento wallet della comunità
/api/dsm/v1/users/	GET	Lista degli utenti
/api/dsm/v1/users/info	GET	Ricerca dettagli utenti per wallet o uuid
/api/dsm/v1/users/{uuid}/communities	GET	Lista comunità con ruolo a cui appartiene l'utente

Tabella 8 – API Marketplace

Route	Method	Description
/api/dsm/v1/community/{_communityId}/models	POST	Assegna modello ad una determinata comunità
/api/dsm/v1/community/{_communityId}/models/{_modelId}	PATCH	Aggiorna parametri del modello per una determinata comunità
/api/dsm/v1/community/{_communityId}/models/{_modelId}/activate	POST	Attiva modello per una determinata comunità
/api/dsm/v1/community/{_communityId}/models/{_modelId}/deactivate	POST	Disattiva modello per una determinata comunità

Tabella 9 – API Associazione Modelli

Route	Method	Description
/api/dsm/v1/config	GET	Recupera tutti i modelli di virtuosismo energetico
/api/dsm/v1/config/{_modelId}	GET	Recupera i parametri di un determinato modello di virtuosismo energetico
/api/dsm/v1/config/{_modelId}	PATCH	Aggiorna i parametri di un determinato modello di virtuosismo energetico
/api/dsm/v1/config/{_modelId}/start	POST	Abilita un determinato modello di virtuosismo energetico
/api/dsm/v1/config/{_modelId}/stop	POST	Disabilita un determinato modello di virtuosismo energetico

Tabella 10 – API Gestione Modelli

Route	Method	Description
/api/dsm/v1/structure/	POST	Recupera tutte le strutture disponibili.
/api/dsm/v1/structure/community/{_communityId}	GET	Recupera strutture disponibili in una comunità specifica.
/api/dsm/v1/structure/owned/{communityId}:	GET	Recupera strutture possedute dall'utente nella comunità corrente.
/api/dsm/v1/structure/{communityId}:	GET	Ottiene dettagli della struttura presente in una determinata comunità.
/api/dsm/v1/structure/{communityId}:	PATCH	Aggiorna dettagli della struttura presente in una determinata comunità.
/api/dsm/v1/structure/{communityId}:	DELETE	Elimina struttura presente in una determinata comunità.
/api/dsm/v1/structure/{structureId}/reservations/{year}/{month}/{day}:	GET	Visualizza prenotazioni giornaliere di una struttura.

Tabella 11 – API Gestione Strutture

Route	Method	Description
/api/dsm/v1/kpi/totalTransactionValue	GET	Mostra il valore totale delle transazioni
/api/dsm/v1/kpi/lessExpensiveTransaction	GET	Restituisce la transazione meno dispendiosa
/api/dsm/v1/kpi/activeUserWallets	GET	Mostra i wallet degli utenti attivi
/api/dsm/v1/kpi/mostExpensiveTransaction	GET	Restituisce la transazione più costosa
/api/dsm/v1/kpi/countActiveUsers	GET	Ritorna il numero totale degli utenti attivi
/api/dsm/v1/kpi/mintedTokens	GET	Restituisce il totale dei token assegnati
/api/dsm/v1/kpi/mostUsedServices	GET	Ritorna i servizi più usati
/api/dsm/v1/kpi/countTransactions	GET	Restituisce il conteggio totale delle transazioni
/api/dsm/v1/kpi/averageTransactionValue	GET	Mostra il valore della media delle transazioni

Tabella 12 – API KPI della Piattaforma

Route	Method	Description
/api/dsm/v1/community/{_communityId}/kpi/countActiveUsers	GET	Ritorna il numero totale degli utenti attivi per la comunità richiesta
/api/dsm/v1/community/{_communityId}/kpi/countTransactions	GET	Restituisce il conteggio totale delle transazioni per la comunità richiesta
/api/dsm/v1/community/{_communityId}/kpi/averageTransactionValue	GET	Mostra il valore della media delle transazioni per la comunità richiesta
/api/dsm/v1/community/{_communityId}/kpi/mostExpensiveTransaction	GET	Restituisce la transazione più costosa per la comunità richiesta
/api/dsm/v1/community/{_communityId}/kpi/mostUsedServices	GET	Ritorna i servizi più usati per la comunità richiesta
/api/dsm/v1/community/{_communityId}/kpi/lessExpensiveTransaction	GET	Restituisce la transazione meno dispendiosa per la comunità richiesta
/api/dsm/v1/community/{_communityId}/kpi/mintedTokens	GET	Restituisce il totale dei token assegnati per la comunità richiesta
/api/dsm/v1/community/{_communityId}/kpi/totalTransactionValue	GET	Mostra il valore della media delle transazioni per la comunità richiesta

/api/dsm/v1/community/{_communityId}/kpi/activeUserWallets	GET	Mostra i wallet degli utenti attivi per la comunità richiesta

Tabella 13 – API KPI della Comunità

6.2.2 Descrizione delle risorse statiche

Nel seguente paragrafo vengono descritte le risorse statiche SQL e Contracts.

Sql: script per la creazione del database DMDB contenente dettagli configurativi dei modelli

Contracts: contiene gli smart contract utilizzati

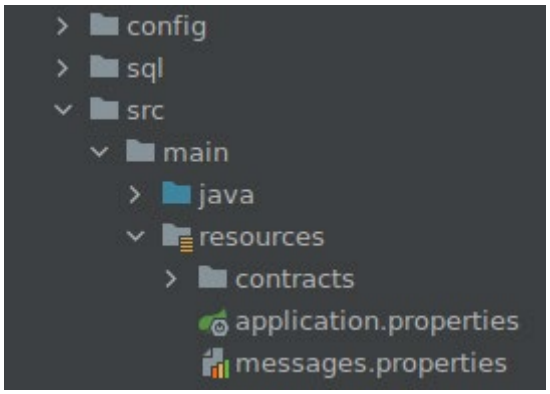

Figura 8 - Struttura delle risorse statiche del progetto

6.2.3 DMDB - Data Model

Nello specifico, sono state realizzate le seguenti tabelle:

- **global_admins:** Tabella che raccoglie le utenze Platform Admin
- **roles:** Tabella che elenca i ruoli
- **service_image:** Tabella che include i dati delle immagini dei servizi nel Marketplace
- **user_community:** Tabella di associazione fra community e UUID utente

Di seguito viene riportata la nuova struttura del DMDB:

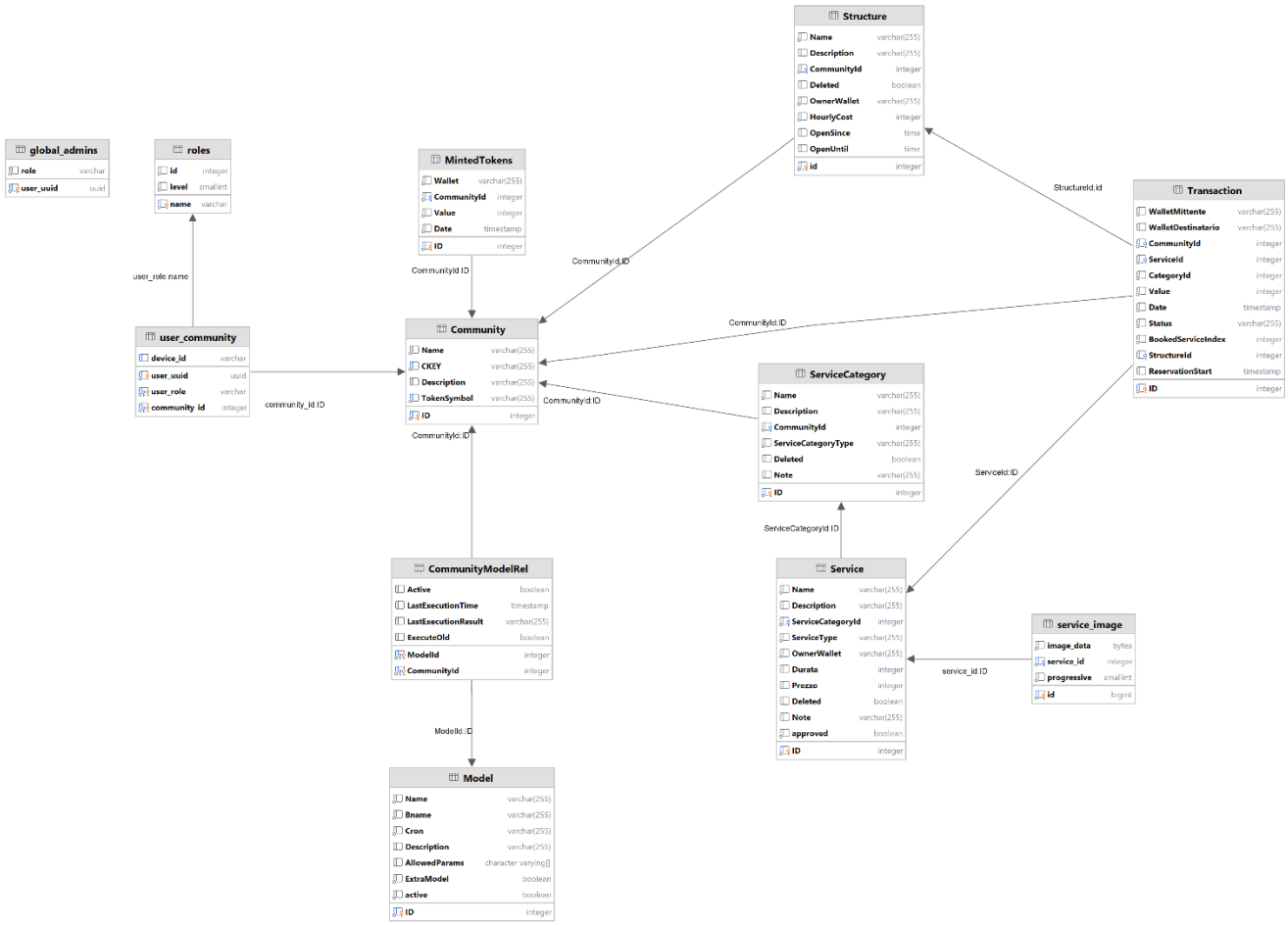

Figura 9 - Modello Logico del DB

6.3 BLOCKCHAIN

La piattaforma è una dApp (Decentralized Application) che interagisce con smart contract distribuiti su una blockchain. Questo consente transazioni sicure, trasparenti e automatizzate senza la necessità di intermediari.

6.3.1 Componenti:

Di seguito vengono descritti tutti i componenti.

- **Ganache**: è una blockchain privata ed emulata localmente, utilizzata per sviluppare, testare e simulare contratti intelligenti in un ambiente sicuro e controllato. Permette agli sviluppatori di eseguire operazioni blockchain senza costi reali e con totale controllo sui parametri della rete, come il tempo e il mining istantaneo delle transazioni. Ganache è molto utilizzato nei progetti Ethereum e compatibili, grazie alla sua facilità di configurazione e agli strumenti per il debugging.
- **Hardhat**: Hardhat è un potente framework di sviluppo per la creazione, il testing e il deployment di smart contract. È dotato di una vasta gamma di strumenti che supportano gli sviluppatori in ogni fase del processo di sviluppo, tra cui:
    - **Gestione automatica del gas**: Hardhat consente di ottimizzare il consumo di gas durante il testing e di monitorare il gas utilizzato da ciascuna transazione, offrendo una visione precisa dei costi.
    - **Simulazione della blockchain locale**: Hardhat Network permette di eseguire una blockchain locale per testare i contratti in un ambiente controllato, con funzionalità come il mining istantaneo e la gestione personalizzata del tempo per simulare scenari complessi.
    - **Debug avanzato**: Include strumenti di debugging che mostrano stack traces dettagliati e informazioni approfondite sugli errori negli smart contract, aiutando a identificare e risolvere rapidamente eventuali problemi.
    - **Supporto per script di automazione**: Grazie alla sua flessibilità, Hardhat permette di scrivere script per il deployment e la gestione degli smart contract, rendendo il processo ripetibile e facilmente adattabile a più ambienti.
- **Metamask**: è un plug-in per browser necessario per interagire con la DApp, esso ha il compito di memorizzare gli account degli utenti (chiave pubblica e chiave privata) e permettere l'interazione dell'account (attualmente selezionato in metamask) con le funzioni esposte dagli smart contract presenti sulla blockchain.

6.3.2 Sviluppo

L'insieme di tutti gli Smart Contract e delle relative interfacce sono stati deployati sulla blockchain Ganache.
Il linguaggio utilizzato per la creazione degli smart contract è Solidity compilato con la versione 0.8.18 di solc-js, gli smart contract sono compatibili con tutte le versioni di solcjs comprese tra 0.7 e 0.9.

6.3.2.1 Migrazione al framework Hardhat

Dal momento che il framework Truffle ha terminato la fase di supporto, si è richiesta la migrazione ad un nuovo framework per la fase di sviluppo, test e deploy degli smart contract sulla blockchain.

È avvenuta, dunque, una migrazione al framework Hardhat.

Il framework Hardhat offre, al pari di Truffle, gli strumenti per:

- compilazione codice sorgente
- deploy degli smart contract su blockchain
- test degli smart contract
- trascodifica del bytecode degli smart contract in javascript

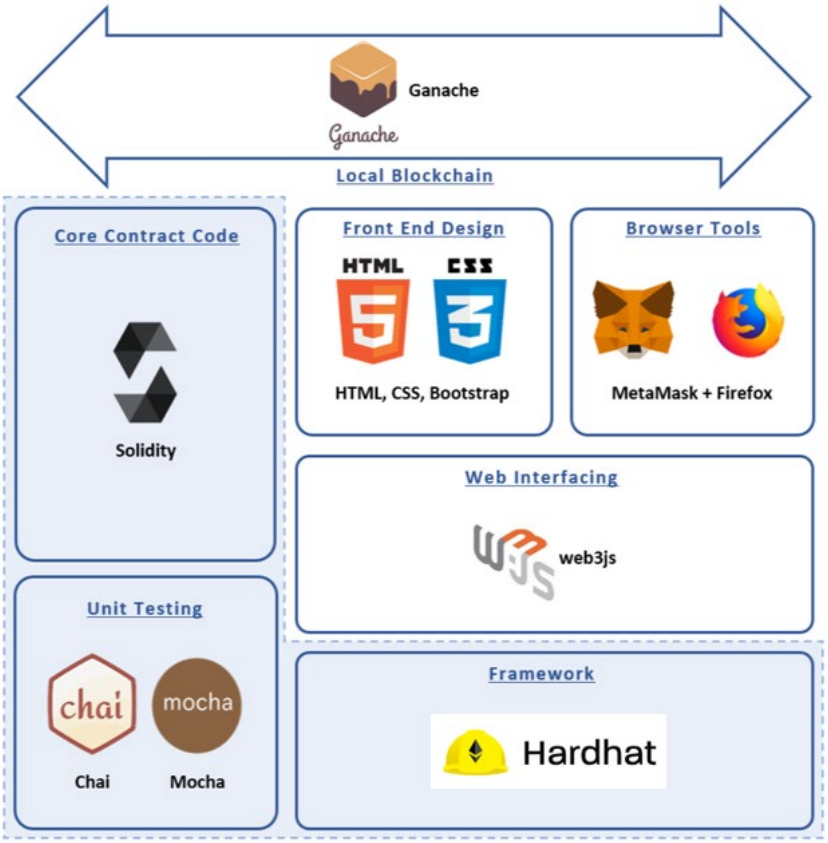

Figura 10 - Framework Hardhat per Deplyment e Testing degli Smart Cotnract

6.3.3 Interfacce e Contratti

Nel seguente paragrafo viene fornita una breve descrizione degli smart contract sviluppati per la piattaforma, vengono, inoltre, evidenziati gli smart contract che sono stati aggiornati al fine di implementare le funzionalità oggetto dell'evolutiva 2024.

- **AccessControl:** Utilizzato per stabilire una gerarchia di ruoli
- **AnnualBonus:** Utilizzato per la gestione del bonus annuale
- **Community:** Utilizzato per la gestione della comunità
- **Demurrage:** Utilizzata per la gestione del demurrage
- **ERC1203:** Utilizzato per la gestione del token di comunità
- **Fee:** Utilizzato per la gestione delle tasse
- **MarketPlace:** Utilizzato per la compravendita di beni e servizi
- **Model:** Utilizzato per la gestione dei modelli di virtuosismo energetico
- **ServiceIncentive:** Utilizzato per la gestione degli incentivi
- **WelcomeBonus:** Utilizzato per la gestione del bonus di benvenuto

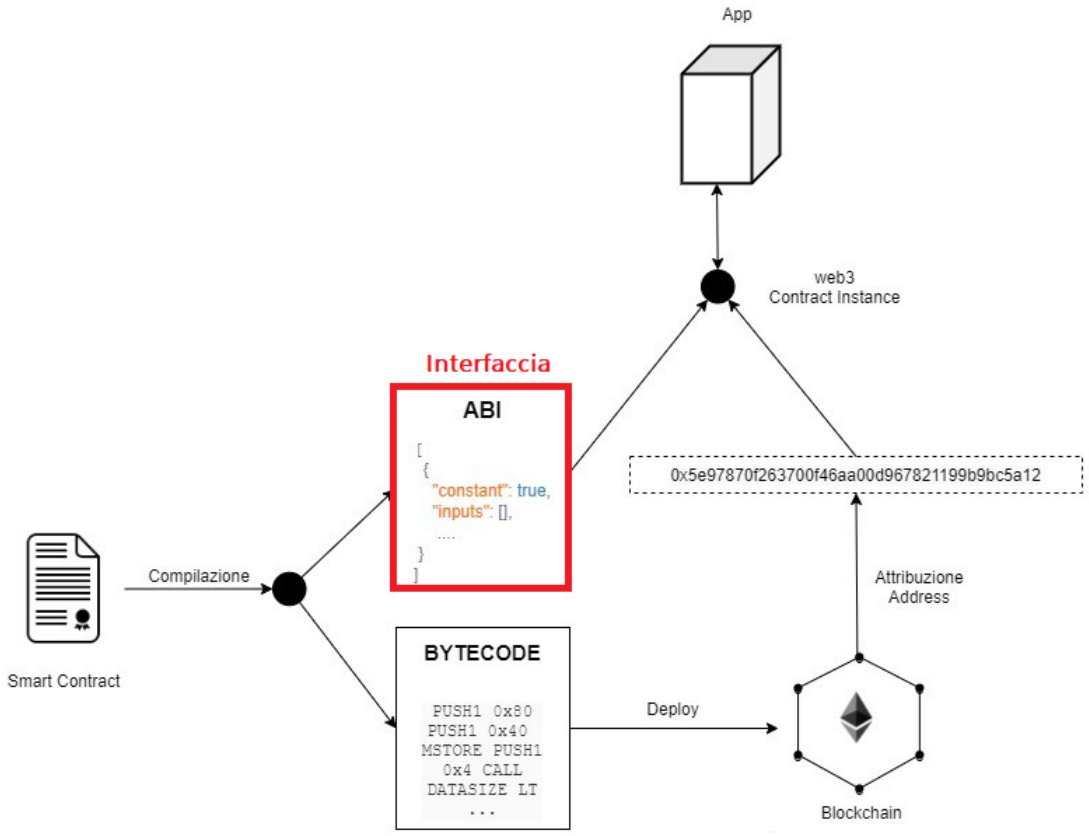

Figura 11 - Schema di deployment e interazione con lo Smart Contract

6.3.4 Struttura del Codice

Nel seguente paragrafo viene descritta la struttura del repository che ospita gli Smart Contract dello Smart Tokenization System.

- **artifacts/contracts**: cartella contenente il bytecode degli gli smart contract compilati
- **contracts**: cartella contenente il codice sorgente di tutti gli smart contract
- **contracts/interfaces**: cartella contenente il codice sorgente di tutte le interfacce degli smart contracts
- **test**: cartella contenente i file di test degli smart contract
- **scripts**: cartella contenente i file utili per poter deployare gli smart contract su blockchain ed eventualmente interagire con essi
- **env**: file contenente le informazioni dinamiche dell'applicazione
- **hardhat.config.js**: file di configurazione di hardhat, in esso vengono definite:
    - la versione di solcjs
    - le stringhe di connessione ad una o più blockchain
    - eventuali parametri di configurazione

6.3.5 Fase di Test

È stata condotta una fase di test su ogni smart contract della piattaforma attraverso l'implementazione di script di test.

Di seguito vengono riportati gli script di test implementati:

Test di verifica

- AccessControl.js

  Script che consente di verificare lo smart contract identificato come "**AccessControl**"
- Community.js

  Script che consente di verificare lo smart contract identificato come "**Community**"
- ERC1203.js

  Script che consente di verificare lo smart contract identificato come "**Erc1203**"
- Marketlace.js

  Script che consente di verificare lo smart contract identificato come "**MarketPlace**"
- Model.js

  Script che consente di verificare lo smart contract identificato come "**Model**"

6.3.6 Fase di Deploy

Di seguito viene riportato il riferimento allo script di deploy

Script di deploy

- 0_deploy.js

  Script che consente di eseguire rilascio degli Smart Contract su una determinata blockchain (scelta tra quelle definite all'interno del file hardhat.config.js)

7 Contributo delle eventuali consulenze alle attività sopra descritte

All'interno del progetto non sono state utilizzate consulenze esterne.

8 Pubblicazioni scientifiche

Non vi sono state pubblicazioni scientifiche risultanti dall'attività svolta

9 Eventi di disseminazione

Non sono scaturiti eventi di disseminazione dall'attività svolta.