

Ricerca di Sistema elettrico



Servizi social per le CER: Creazione di modelli e strumenti software per l'adesione a una comunità energetica e per lo scambio e acquisto di energia fra utenti della comunità in ambito
LTE (LA1.18)

Francesco Bruschi

Servizi social per le CER: Creazione di modelli e strumenti software per l'adesione a una comunità energetica e per lo scambio e acquisto di energia fra utenti della comunità in ambito LTE

LA 1.18 - Servizi social per le CER: Creazione di modelli e strumenti software per l'adesione a una comunità energetica e per lo scambio e acquisto di energia fra utenti della comunità in ambito LTE

F. Bruschi (DEIB - Politecnico di Milano)

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: LA 1.18

Responsabile del Progetto: Claudia Meloni, ENEA

Responsabile del Work Package: Angelo Frascella, ENEA

Responsabile Linea di Attività: Francesco Bruschi, Dipartimento di Elettronica, Informazione e Bioingegneria - Politecnico di Milano

Mese inizio previsto: 13

Mese inizio effettivo: 13

Mese fine previsto: 36

Mese fine effettivo: 36

Il presente documento descrive le attività di ricerca svolte all'interno dell'Accordo di collaborazione: "Creazione di modelli e strumenti software per l'adesione a una comunità energetica e per lo scambio e acquisto di energia fra utenti della comunità"

Indice

1	Risultati attesi.....	5
2	Risultati ottenuti.....	6
3	Prodotti attesi.....	8
4	Prodotti sviluppati.....	9
5	Analisi degli scostamenti su attività e risultati.....	10
6	Sintesi delle attività svolte.....	11
7	Dettaglio delle attività svolte.....	12
8	Contributo delle eventuali consulenze alle attività sopra descritte.....	18
9	Pubblicazioni scientifiche.....	19
10	Eventi di disseminazione.....	20

Indice delle figure

Figura 1 - Apertura delle buste.....	13
Figura 2 - saldi finali con transazioni in stato pending.....	14
Figura 3 - fasi in cui si articola il protocollo di mercato.....	14

1 Risultati attesi

Lista dei risultati attesi come da capitolato vigente

- 1.** Studio e analisi dei modelli necessari e del progetto di un dimostratore per l'attivazione e la gestione di procedure di acquisto e vendita (o scambio) di energia fra utenza appartenenti a comunità energetiche facenti parte di una local token economy, e sviluppo del relativo software dimostratore.
- 2.** Studio e analisi dei modelli necessari e del progetto dell'applicazione per la gestione di procedure di attivazione di cittadini verso la local token economy, e sviluppo del relativo software dimostratore.
- 3.** Dimostratore software basato su tecnologia Blockchain e Smart Contract per la sottoscrizione, attraverso un ambiente marketplace, di utenti alla partecipazione a comunità energetiche facenti parte di una local token economy e la gestione di transazioni relative a procedure di acquisto e vendita (o scambio) di energia, che implementi i modelli messi a punto.

2 Risultati ottenuti

Lista dei risultati ottenuti:

- Studio e analisi dei modelli necessari e del progetto di un dimostratore per l'attivazione e la gestione di procedure di acquisto e vendita di energia fra utenze appartenenti a comunità energetiche e partecipanti ad un mercato locale dell'energia, e sviluppo del relativo software dimostratore basato su tecnologia Blockchain e Smart Contract.

Il dimostratore copre i seguenti aspetti:

1. Sviluppo di una piattaforma decentralizzata per il trading energetico
 - Realizzazione di un prototipo funzionante che dimostra la fattibilità tecnica
 - Beneficio: Abilitazione di scambi diretti tra i membri registrati della comunità energetica, senza intervento di intermediari
2. Implementazione di un meccanismo di mercato trasparente
 - Sistema basato su blockchain che garantisce la tracciabilità delle transazioni
 - Beneficio: Maggiore fiducia tra i partecipanti e riduzione delle possibilità di manipolazione
3. Automazione del processo di trading
 - Sviluppo di smart contracts per la gestione automatizzata degli ordini e pagamenti
 - Beneficio: Riduzione dei costi operativi e dei tempi di esecuzione delle transazioni
4. Validazione del concetto di comunità energetica decentralizzata
 - Dimostrazione della possibilità di gestire scambi energetici peer-to-peer
 - Beneficio: Base per l'espansione delle comunità energetiche in Italia

Come risultati abbiamo inoltre:

5. Attività di supporto (1-2 giorni):
 - Coaching relativo allo sviluppo e all'uso del software sviluppato
6. Report documentale, sviluppato nel corso della collaborazione, contenente:
 - Descrizione e analisi dei modelli di acquisto/vendita energia
 - Documentazione completa del primo dimostratore (progettazione, sviluppo, installazione, utilizzo)
 - Descrizione e analisi dei modelli per l'adesione alle comunità
 - Documentazione completa del secondo dimostratore

- Studio e analisi dei modelli necessari e del progetto dell'applicazione per la gestione di procedure di attivazione di cittadini, e sviluppo del relativo software dimostratore: implementazione di un servizio di registrazione per l'abilitazione degli utenti alla partecipazione nel mercato energetico locale.

3 Prodotti attesi

Produzione di un dimostratore software basato su tecnologia Blockchain e Smart Contract per consentire agli utenti di una comunità energetica di sottoscrivere per la partecipazione ad un mercato energetico, e per operare la gestione, sempre attraverso degli opportuni smart contract, del processo di acquisto e vendita di energia, e delle relative transazioni di valore, che implementi il modello individuato precedentemente.

Lista dei prodotti hardware/software eventualmente attesi per la LA, secondo quanto stipulato nell'accordo di collaborazione:

7. Software Dimostratore per Trading Energetico
 - a. Implementazione in Solidity degli smart contracts
 - b. Interfaccia utente in framework Angular
 - c. Sistema per la gestione delle transazioni di acquisto/vendita energia
 - d. Componenti per il calcolo del prezzo di clearing
 - e. Sistema per la validazione delle transazioni energetiche
8. Software Dimostratore per Gestione Adesioni
 - a. Smart contracts per la gestione dei contratti di adesione
 - b. Interfaccia web per l'onboarding degli utenti
 - c. Sistema di verifica e validazione delle adesioni
 - d. Componenti per l'amministrazione delle utenze
9. Componenti di Integrazione
 - a. Interfacce per l'integrazione con la piattaforma esistente
 - b. Componenti per la gestione dei dati esterni alla blockchain
 - c. API per lo scambio dati tra i sistemi

Nello sviluppo del software si richiede l'adozione di alcune specifiche tecniche:

- Linguaggio smart contracts: Solidity
- Database: PostgreSQL o MongoDB (se necessario)
- Frontend: Framework Angular
- Backend: Java, Javascript e/o Python
- Utilizzo di librerie open-source
- Compatibilità con l'attuale piattaforma per Local Energy Community

4 Prodotti sviluppati

Durante le attività è stato sviluppato un dimostratore software basato su tecnologia blockchain, costituito da diversi componenti, elencati successivamente:

10. Smart contracts:
 - SessionHandler.sol: contratto che gestisce le sessioni di mercato
 - MarketplaceContract.sol: contratto che gestisce gli ordini, i fondi degli utenti e gli scambi di energia tokenizzata. Permette inoltre di notarizzare il contratto di adesione dell'utente alla comunità energetica effettuandone di fatto la registrazione nel sistema
11. Frontend: applicazione Angular che permette agli utenti di interagire con il sistema attraverso un wallet compatibile con WalletConnect (es. MetaMask)
12. Componente off-chain (Blackbox): servizio Flask che simula le logiche off-chain, ossia procedura di clearing price e rimodulazione degli ordini in cui possono essere implementati i meccanismi effettivi in una fase successiva
13. Script di deploy: progetto Hardhat contenente gli script necessari per automatizzare il deploy degli smart contract sulla rete locale Hyperledger Besu utilizzata nel prototipo
14. Infrastruttura Docker: organizzazione delle componenti frontend, blackbox, deploy e di un nodo Hyperledger Besu locale all'interno di container dedicati per facilitare la gestione delle varie parti del sistema

Il progetto del software sviluppato è disponibile al seguente link:

<https://github.com/bishub-polimi/enea-energy-marketplace>

da dove è possibile scaricare tutti i codici dei vari componenti e le istruzioni per eseguirlo.

5 Analisi degli scostamenti su attività e risultati

Nello sviluppo del progetto non si sono verificati scostamenti significativi né dal punto di vista tecnico né economico. La pianificazione accurata delle varie fasi, la collaborazione e la comunicazione efficace tra le parti hanno permesso di raggiungere risultati conformi alle aspettative senza riscontrare criticità significative.

6 Sintesi delle attività svolte

15. Fase di analisi:

- Studio del contesto delle comunità energetiche
- Identificazione delle sfide principali (trasparenza, non-manomissibilità, disintermediazione)
- Analisi dei requisiti funzionali del sistema

16. Fase di design:

- Definizione dell'architettura del sistema
- Progettazione del protocollo di trading su blockchain
- Scelta dello stack tecnologico

17. Fase implementativa:

- Sviluppo degli smart contracts
- Implementazione del frontend in Angular
- Creazione dei servizi off-chain in Flask
- Integrazione dei componenti

18. Fase di testing:

- Testing degli smart contracts
- Validazione del sistema integrato
- Verifica dei flussi di trading

19. Documentazione:

- Stesura della documentazione tecnica
- Creazione della guida all'utilizzo
- Redazione del report finale

7 Dettaglio delle attività svolte

Attività svolte e dei risultati ottenuti in relazione ai risultati attesi.

1. **Obiettivo del progetto.** Lo scopo principale del progetto è stato lo sviluppo di un meccanismo di mercato dell'energia tra attori di una comunità energetica. Le comunità energetiche locali rappresentano un modello rivoluzionario di produzione e consumo energetico, introdotte in Italia in seguito all'approvazione del Decreto Milleproroghe 162/2019, previsto dalla Direttiva Europea RED II (2018/2001/UE). Secondo la ricerca condotta dal Politecnico di Milano (Electricity Market Report), entro il 2025 si prevede che le energy community in Italia raggiungeranno circa 40.000 unità, coinvolgendo 1,2 milioni di famiglie, 200.000 uffici e 10.000 PMI. La collaborazione tra ENEA, Politecnico di Milano e Politecnico di Torino ha permesso di studiare e implementare un sistema che sfrutta algoritmi specifici per stabilire il prezzo di clearing e la tecnologia blockchain per gestire gli ordini di compravendita e la liquidazione dei pagamenti.

2. **Architettura del sistema.** È stata implementata un'infrastruttura basata su Docker, composta da quattro container principali:
 - a. **besu:** Un nodo blockchain locale Hyperledger Besu che supporta diversi meccanismi di consenso. Nel contesto del PoC è stato utilizzato l'algoritmo PoW. La rete è configurata come "gas-free network" (gasPrice = 0), eliminando le commissioni per le transazioni.
 - b. **deploy:** Un progetto Hardhat che effettua il deploy dei contratti sulla rete locale. Hardhat è stato scelto per la sua integrazione nativa con TypeScript e gli strumenti avanzati per il testing.
 - c. **frontend:** Un'applicazione Angular che offre un'interfaccia utente, strutturata in componenti, servizi e moduli.
 - d. **blackbox:** Un servizio Flask che implementa le componenti off-chain del sistema, in particolare l'algoritmo di clearing price.

I container comunicano attraverso una rete custom denominata *enea-demo*, mentre i file condivisi sono gestiti attraverso due volumi:

shared-volume: Contiene il file `.env` con le variabili d'ambiente

shared-artifacts: Contiene gli artefatti della compilazione dei contratti

3. **Sviluppo degli smart contract.** Sono stati sviluppati due smart contract principali:
 - a. **SessionHandler.sol:** Gestisce le sessioni di mercato, definendo le scadenze e verificando lo stato delle sessioni. Implementa funzionalità come:
 - Apertura di nuove sessioni (`newSession`)

- Gestione delle scadenze (getCommitDeadline, getRevealDeadline)
- Verifica dello stato di validazione (setPriorSessionValidation)

b. **MarketplaceContract.sol**: Gestisce gli ordini, i fondi degli utenti e le transazioni. Include funzioni per:

- Registrazione degli utenti (registerUser)
- Sottomissione e rivelazione degli ordini (submitCommitments, revealOrders)
- Gestione dei fondi (checkFunds, claimAvailableFunds, withdrawAllFunds)
- Elaborazione degli ordini e delle transazioni (setFinalOrders, setValidTransactions, execute)

Entrambi i contratti sono stati sviluppati in Solidity e sono compatibili con piattaforme EVM, facilitando la loro migrazione futura su altre reti.

4. Implementazione del protocollo di mercato. Il protocollo implementato comprende le seguenti fasi:

a. **Apertura di una nuova sessione di mercato**: Il Gestore del Mercato avvia una sessione definendo le scadenze.

b. **Sottomissione degli ordini in busta chiusa (commitment)**: Gli utenti inviano hash dei loro ordini insieme a una caparra entro l'ora t_c .

c. **Rivelazione degli ordini (reveal)**: Gli utenti rivelano i dettagli degli ordini entro l'ora $t_r > t_c$, con verifica della corrispondenza con i commitment inviati precedentemente.

```

REVEAL SUBMISSIONS:
-----
Current time: 15/12/2023, 00:24:13
-----
Reveal Event:
  User: 0x70997970C51812dc3A010C7d01b50e0d17dc79C8
  Session ID: 1
  Funds: 29970
  revealedOrders: true,0x70997970C51812dc3A010C7d01b50e0d17dc79C8,0,100,50,false,true,0x70997970C51812dc3A010C7d01b50e0d17dc79C8,0,200,50,false,true,0x70997970C51812dc3A010C7d01b50e0d17dc79C8,0,300,50,false
Reveal Event:
  User: 0x3C44CdD86a900fa2b585dd299e03d12FA4293BC
  Session ID: 1
  Funds: 8970
  revealedOrders: true,0x3C44CdD86a900fa2b585dd299e03d12FA4293BC,1,100,50,false,true,0x3C44CdD86a900fa2b585dd299e03d12FA4293BC,1,200,50,false,true,0x3C44CdD86a900fa2b585dd299e03d12FA4293BC,1,300,50,false
-----
Funds after the off-chain algorithm:
-----
User: 0x70997970C51812dc3A010C7d01b50e0d17dc79C8
Locked funds in the contract: 29970
Available funds in the contract: 0
Personal wallet funds: 19999970020
-----
User: 0x3C44CdD86a900fa2b585dd299e03d12FA4293BC
Locked funds in the contract: 8970
Available funds in the contract: 0
Personal wallet funds: 19999991020
-----

```

Figura 1 - Apertura delle buste

d. **Calcolo del clearing price (off-chain)**: Il Gestore del Mercato utilizza un algoritmo off-chain per determinare il prezzo di clearing e rimodulare gli ordini.

e. **Notarizzazione degli ordini rimodulati**: Gli ordini rielaborati vengono caricati on-chain tramite il metodo setFinalOrders.

f. **Validazione degli ordini:** Si confrontano gli ordini con i dati reali di consumo/produzione attraverso un'elaborazione off-chain.

g. **Esecuzione delle transazioni validate:** Gli utenti possono avviare l'esecuzione degli ordini e ricevere i fondi dovuti.

```

ORDERS EXECUTION:

Final balances:

User: 0x70997970C51812dc3A010C7d01b50e0d17dc79C8

Locked funds in the contract: 27000
Available funds in the contract: 0
Personal wallet funds: 19999972990

User: 0x3C44CdDdB6a900fa2b585dd299e03d12FA4293BC

Locked funds in the contract: 8100
Available funds in the contract: 0
Personal wallet funds: 19999991890
  
```

Figura 2 - saldi finali con transazioni in stato pending

Come illustrato nella figura successiva, il protocollo si articola in una fase di sottomissione (rossa), una fase di rimodulazione (arancione) e una fase di validazione/esecuzione (verde).

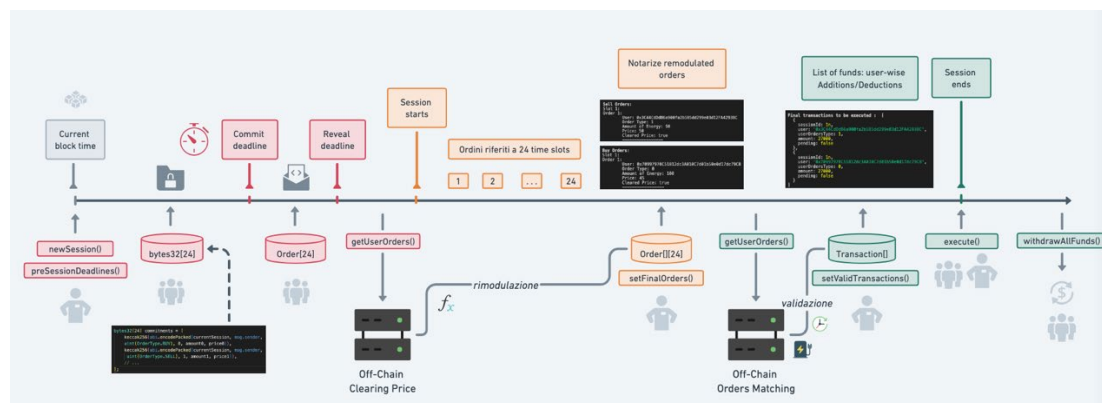


Figura 3 - fasi in cui si articola il protocollo di mercato

5. Gestione dei fondi. È stato implementato un sistema di gestione dei fondi basato su token ERC-20, simulando l'uso di una stablecoin. Dopo un'analisi delle alternative, si è optato per l'utilizzo di una stablecoin esistente su rete permissionless, che offre:

- Riduzione dell'effort di sviluppo e gestione
- Interoperabilità con altre applicazioni
- Eliminazione della necessità di operare come custode di valore

Il sistema implementa un "conto" address-based all'interno del contratto principale, permettendo:

- Deposito e prelievo di fondi
 - Blocco di fondi a garanzia degli ordini
 - Restituzione automatica di fondi non utilizzati
 - Gestione delle penalità per comportamenti scorretti
6. Interfaccia utente. È stata sviluppata un'applicazione web Angular che permette agli utenti di:
- a. **Registrarsi alla comunità energetica:** Caricando il contratto firmato e calcolandone l'hash per la verifica on-chain.
 - b. **Sottomettere e rivelare ordini:** Attraverso moduli dedicati e funzioni di interazione con gli smart contract.
 - c. **Visualizzare il proprio bilancio:** Monitorando i fondi bloccati e disponibili.
 - d. **Eeguire le transazioni validate:** Avviando il processo di liquidazione.
 - e. **Prelevare i fondi disponibili:** Ritirando i fondi non utilizzati o guadagnati.

L'applicazione è organizzata in componenti come:

- ConnectButton: Per la gestione della connessione a Ethereum
 - SubmitCommitments: Per l'invio e la rivelazione dei commitment
 - GetOrdersButton: Per l'estrazione e la notarizzazione degli ordini
 - SettleComponent: Per l'esecuzione degli ordini e il prelievo dei fondi
7. Integrazione con il wallet Metamask. Questa integrazione è stata funzionale alla:
- Gestione sicura delle chiavi private
 - Firma delle transazioni
 - Connessione a reti blockchain configurabili
 - Visualizzazione dei saldi in token ERC-20
- L'interazione avviene tramite la libreria ethers.js, che fornisce:
- BrowserProvider: Per interagire con wallet compatibili con EIP-1193
 - Signer: Per gestire la firma delle transazioni

8. Testing. Sono stati implementati test automatizzati utilizzando Hardhat e Chai per verificare:
- Funzionamento degli smart contract
 - Corretta esecuzione del protocollo di mercato
 - Gestione appropriata dei fondi
 - Interazione tra acquirenti e venditori

I test simulano un flusso completo di una sessione di mercato, includendo tutte le fasi dal commitment all'esecuzione finale.

9. Servizi off-chain. Sono stati sviluppati servizi off-chain a scopo dimostrativo nel container "blackbox" per:
- a. **Calcolo del clearing price:** Algoritmo che determina il prezzo di liquidazione e rimodula le offerte per una distribuzione equa delle risorse.

b. **Validazione degli ordini:** Confronto con i dati reali di consumo/produzione per verificare il rispetto degli accordi.

Questi servizi off-chain sono stati implementati in Python e resi disponibili attraverso un'API Flask, consentendo una separazione efficace tra la logica computazionalmente intensiva e l'esecuzione on-chain.

10. Risultati ottenuti:

a. **Prototipo funzionante:** È stato sviluppato un prototipo completo di un mercato energetico basato su blockchain, dimostrando la fattibilità dell'approccio.

b. **Protocollo sicuro:** Il sistema garantisce la segretezza delle offerte e previene manipolazioni attraverso l'uso di commitment.

c. **Gestione flessibile dei fondi:** Gli utenti possono depositare, prelevare e gestire i propri fondi in modo sicuro.

d. **Integrazione on-chain/off-chain:** È stato raggiunto un equilibrio efficace tra operazioni on-chain (trasparenti e immutabili) e off-chain (efficienti e scalabili).

e. **Interfaccia utente intuitiva:** L'applicazione web semplifica l'interazione con la blockchain anche per utenti non esperti.

f. **Test completi:** Una suite di test verifica il corretto funzionamento di tutte le componenti del sistema.

11. Aspetti innovativi:

a. **Uso di commitment:** Garantisce la segretezza degli ordini prima della fase di reveal, prevenendo manipolazioni del mercato.

b. **Sistema di penalità:** Scoraggia comportamenti scorretti attraverso la requisizione della caparra in caso di inadempienza.

c. **Algoritmi off-chain:** Ottimizza l'uso delle risorse blockchain spostando i calcoli computazionalmente intensivi off-chain.

d. **Utilizzo di token ERC-20:** Semplifica la gestione dei fondi e garantisce interoperabilità con l'ecosistema DeFi.

12. Sfide affrontate e risolte:

a. **Sincronizzazione on-chain/off-chain:** È stata implementata una soluzione robusta per la comunicazione tra componenti on-chain e off-chain.

b. **Gestione sicura dei fondi:** Il sistema previene la perdita di fondi anche in caso di errori o comportamenti malevoli.

c. **Usabilità dell'interfaccia:** Sono state superate le difficoltà di creazione di un'interfaccia semplice per interazioni complesse con la blockchain.

d. **Ambiente di test realistico:** È stato sviluppato un ambiente che simula fedelmente le condizioni di un mercato energetico reale.

Questa implementazione fornisce una base solida per lo sviluppo futuro di mercati energetici decentralizzati, dimostrando la fattibilità e i potenziali vantaggi dell'utilizzo della tecnologia blockchain in questo contesto.

8 Contributo delle eventuali consulenze alle attività sopra descritte

Nessuna consulenza usata nella LA

9 Pubblicazioni scientifiche

Nessuna pubblicazione scientifica scaturita dalla LA

10 Eventi di disseminazione

Nessun evento di disseminazione scaturito dall'attività svolta