UNIVERSITÀ DEGLI STUDI DI CATANIA

Dipartimento di Matematica e Informatica Corso di Laurea Triennale in Informatica

Ernesto Casablanca

Decentralizzazione del mercato dell'energia

Decentralization of the energy market

Relazione Progetto Finale

Relatore: Prof. Giuseppe Pappalardo
Correlatore: Dott. Giovanni Marotta

Anno Accademico 2020 - 2021

Indice

Abstract

Il mondo della produzione e distribuzione dell'energia elettrica sta andando incontro ad un processo di decentralizzazione sempre più rapido.
La spinta nasce dall'ingresso nel mercato di nuovi piccoli produttori di energia rinnovabile con capacità variabili e da una domanda che necessità di sempre più flessibilità e con una rinnovata sensibilità ambientale.
Per poter supportare questo nuovo tipo di mercato, incentrato su un rapporto più diretto fra produttore e consumatore, diversi progetti hanno preso in considerazione la tecnologia delle blockchain, sfruttandone i punti di forza e cercando di superarne le limitazioni.
In questo documento ci si concentrerà particolarmente sulle soluzioni in questo ambito proposte dal progetto Energy Web, al fine di fornire un esempio concreto di una possibile implementazione.
I concetti chiave possono comunque essere applicati ad altri progetti della stessa natura.

Capitolo 1

Introduzione

1.1 Il mercato dell'energia

La gestione centralizzata della rete elettrica è stata a lungo la scelta più ragionevole, ma adesso ha senso rivedere questo paradigma. In passato, i grandi operatori elettrici erano anche quelli che producevano anche la quasi totalità dell'energia, gli unici ad avere a disposizione i mezzi per realizzare impianti costosi e complessi. Ad effettuarne poi la distribuzione sono poi i Transmission System Operator (TSO), spesso con coperture nazionali, come la italiana Terna [1].
La situazione odierna, tuttavia, è diversa. Negli anni abbiamo assistito ad una riduzione progressiva del costo degli impianti per la produzione di energia rinnovabile, anche grazie a regolamentazioni estremamente favorevoli [2].
La conseguenza è una diffusione sempre più capillare di piccoli produttori di energia elettrica, che attraverso piccoli impianti, spesso domestici, sono in grado di offrire un contributo attivo alla rete [3].
Ad essi si aggiungono dispositivi come veicoli elettrici e nuovi sistemi di stoccaggio di energia, i quali, se ben gestiti, possono offrire un servizio utile all'intera rete elettrica.
Tutti questi dispositivi, definiti Distributed Energy Resource (DER), introducono la necessità di un flusso bidirezionale di energia che li includa come agenti attivi [4].
Inoltre, la natura variabile e difficile da prevedere della maggior parte delle fonti di energia rinnovabile o Renewable Energy Sources (RES) ne richiede una gestione flessibile, in grado di garantire robustezza e stabilità all'intera rete [5].

La gestione di queste piccole realtà, mal si sposa con l'attuale struttura

centralizzata, e impone un cambio di paradigma verso un'architettura distribuita, in grado di includere i DER e mettere a più stretto contatto produttore e consumatore.

Figura 1.1: Da rete da centralizzata a rete distribuita [6]

1.2 La tecnologia delle blockchain

La tecnologia delle blockchain esiste già da parecchi anni, ma negli ultimi tempi ha ricevuto crescenti attenzioni da studiosi e industrie, intenzionate a sfruttarne a pieno il potenziale.
Tralasciando i dettagli più tecnici della tecnologia, che è possibile approfondire in pubblicazioni come quelle referenziate in [7] e [8], la blockchain può essere vista come una lista di transazioni pubblica, decentralizzata e immutabile, se non per l'aggiunta di nuove transazioni, della quale chiunque può possedere una copia.
L'integrità e la correttezza delle transazioni vengono garantite dagli algoritmi crittografici, che rendono molto facile verificare la validità della transazione. Ciò permette l'interazione di una qualsiasi entità con le altre senza che questa debba fare affidamento su nient'altro che l'algoritmo di consenso distribuito che governa la blockchain, eliminando la necessità di un intermediario fra le parti. Il risultato è una rete distribuita Peer to Peer (P2P) estremamente robusta che si presta a molteplici applicazioni.
A rendere ancora più appetibile questa tecnologia sono gli smart contracts, presenti in reti come Ethereum [9]. Semplificando, si tratta di veri e propri software rilasciati sulla blockchain in grado di compiere svariate azioni in maniera automatica seguendo la loro programmazione.

Capitolo 2

Energy web

2.1 Cos'è Energy Web

Energy Web (EW) è un progetto nato nel 2017 con sede in Zugo, Svizzera [10]. Ad affiancarlo e dargli valore vi sono molti partner commerciali, comprese aziende molto conosciute nel settore energetico [11].
L'obiettivo prefissato da EW è quello di spingere per uno sviluppo del settore che punti verso un abbassamento delle emissioni di carbonio e che sia in grado di gestire la decentralizzazione del mercato delle risorse. Per farlo, EW utilizza tecnologie distribuite e open-source dalle quali partire per realizzare un'infrastruttura commerciale specifica per l'ambito energetico [12].
Nel 2019, EW ha rilasciato Energy Web Chain (EWC), una blockchain pubblica basata su Ethereum, sulla quale si basa l'intero ecosistema di EW: Energy Web Decentralized Operating System (EW-DOS).

2.2 EW-DOS

Il cuore del progetto di EW è EW-DOS, un'infrastruttura digitale open-source ad accesso pubblico e decentralizzato.
L'idea è quella di fornire un insieme di strumenti e servizi che rendano il più semplice possibile lo sviluppo di Decentralized Application (DApp) mirate al settore energetico, anche se, ovviamente, le varie tecnologie possono essere applicate anche ad ambiti più generici.
L'intero sistema è stato pensato come tre strati sovrapposti, in cui ogni strato si basa su quello sottostante per implementare ulteriori funzionalità, come mostrato in Fig. 2.1.

I tre strati sono:

- Trust - EWC
- Utility - Servizi e astrazioni sopra la blockchain
- Toolkit - Frameworks e toolkit per la costruzione di applicazioni

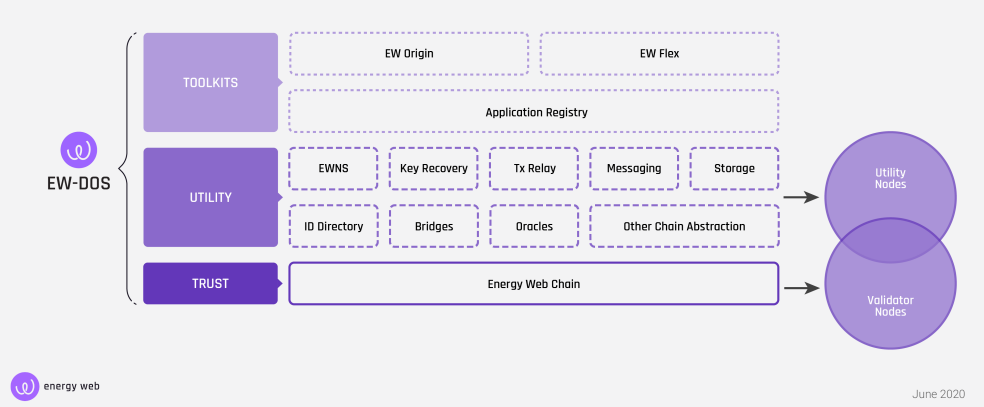


Figura 2.1: Visualizzazione della struttura di EW-DOS [13]

2.3 Trust

Il livello Trust comprende la EWC (cfr. Cap. 3), il cui ruolo principale è quello di assicurare che ci sia consenso sui dati e che tutte le applicazioni e gli smart contracts si comportino in maniera deterministica.
Si tratta di una blockchain basata su Ethereum, Ethereum Virtual Machine (EVM) inclusa, e rispetta tutti gli Ethereum Request for Comment (ERC).
Il token digitale per il pagamento delle transazioni e dei servizi offerti dalla piattaforma è l'Energy Web Token (EWT) e l'algoritmo di consenso è il Proof of Authority (PoA).
È presente anche una test-net, chiamata Volta, usata per testare i progetti e le applicazioni prima di lanciarle sulla main-net.

2.4 Utility

Il livello di utility è composto da un insieme di servizi basati sulla EWC che hanno lo scopo di rendere quanto più accessibile e invitante possibile l'intera infrastruttura per gli sviluppatori di DApp, ed inoltre mettendo a disposizione un gran numero di smart contracts per il backend e di librerie

per il frontend [14].
Come anticipato, il pagamento di servizi del livello di Utility della piattaforma vengono pagati in EWT.
I nodi della blockchain che forniscono queste funzionalità sono chiamati Utility Nodes, e vengono ricompensati attraverso il meccanismo dello Staking (cfr. sez. 3.8).

È possibile individuare tre ampie categorie di servizi:

- Esperienza dell'utente finale
- Interoperabilità Multipiattaforma
- Performance delle applicazioni

2.4.1 Esperienza dell'utente finale

- Energy Web Name Service (EWNS) - Similmente ad un DNS, associa domini arbitrari ad indirizzi sulla blockchain
- DID Key Recovery - Permette il recupero della propria DID (cfr. sez. 4.2) nel caso si fosse persa la chiave privata
- Transaction Relay - Servizio che fa da intermediario fra un cliente e la blockchain, nascondendo l'utilizzo della stessa e simulando una più tradizionale interazione client-server

2.4.2 Interoperabilità Multipiattaforma

- Bridges - Consente il trasferimento di token fra blockchain diverse. Attualmente è utilizzato per collegare EWC con Ethereum
- Oracles - Particolari nodi che è possibile consultare con il protocollo Chainlink per ricevere dati di eventi esterni alla blockchain [15] [16]

2.4.3 Performance delle applicazioni

- Identity Directory - Smart contract che contiene tutti le Decentralized IDentifiers (DID)
- Messaging - Un sistema di messaggistica che sfrutta le DID per assicurare validità ed autenticità dei messaggi

- Storage - L'utilizzo di sistemi di storage distribuiti esterni, come IPFS [17] rende possibile limitare al minimo indispensabile la quantità di dati salvati sulla blockchain

2.5 Toolkit

Il livello toolkit comprende una serie di framework ed applicazioni di esempio utili per realizzare DApp che sfruttino al massimo le funzionalità di EW-DOS [14]. Sebbene siano pensati per il settore energetico, la loro natura open-source li rende ottime basi di partenza per costruire soluzioni anche per altri ambiti.

2.5.1 Application Registry

Architettura in grado creare registri di DID che verifichino che soddisfino la condizione specificata, come ad esempio la località geografica o il possesso di una certa qualifica.

Questo registro sarà poi utilizzato da una DApp per creare un servizio di autorizzazione e autenticazione.

Ogni DApp deve fare riferimento ad un Application Registry, che può essere riutilizzato in più applicazioni.

2.5.2 EW origin

Framework per sviluppare applicazioni che supportino il tracciamento, la trasmissione e il conferimento di Energy Attribute Certificate (EAC) alle RES secondo gli standard del settore.

I due attori principali sono il Registry e l'Issuer. Il primo salva e amministra le informazioni legate ad utenti e DER, con la possibilità di mantenere dati potenzialmente sensibili off-chain, cioè utilizzando dello storage al di fuori della blockchain. Il secondo potrà essere utilizzato dalle autorità competenti per coniare nuovi EAC che siano tracciabili, con una implementazione basata sullo standard ERC-1155.

2.5.3 EW flex

Software open source attualmente in sviluppo per la gestione dei DER e le operazioni che li coinvolgono, permettendone una facile connessione alla rete e sottoponendo le offerte ad un operatore autorizzato [14].

Sarà composto dai seguenti moduli:

- Flex Nodes - Cluster di nodi che eseguono materialmente la business logic che gestisce offerta e domanda, e le accoppia secondo quanto stabilito dall'operatore responsabile.
- Flex Clients - Insieme di tutti i DER che partecipano al mercato dell'energia, inclusi dispositivi Internet of Things (IoT)
- Flex Bridge - Modulo che permette la comunicazione e la coordinazione con gli operatori della rete elettrica
- Flex Governance - Serie di smart contracts che governano i Flex Nodes. Permettono agli enti predisposti di imporre le regolamentazioni vigenti

Capitolo 3

Energy Web chain

3.1 Confronto con Ethereum

Non è un mistero che la EWC sia basata su Ethereum, più precisamente sulla sua ottava hard-fork, Istanbul [18]. È in programma una ulteriore migrazione alla hard-fork Berlino, che avverrà quanto prima [19].
Anche alcuni dei servizi offerti, come l'EWNS sono fork con modifiche minime di progetti nati su Ethereum.

Tuttavia, ci sono alcuni aspetti in cui le due reti differiscono, come riassunto nella segunte tabella:

Problema	Modifica
Basso throughput, alti costi, scalabilità limitata	Utilizzo della PoA: incrementa il throughput fino a 30x
Poco adatta a piccoli dispositivi (IoT)	Maggiore focus sui **light client** per connettere anche piccoli dispositivi (IoT)
Nessuna distinzione fra i nodi con diverse autorizzazioni	È possibile differenziare fra nodi con compiti ed autorizzazioni diverse
Difficoltà a gestire transazioni che necessitino di privacy	Possibilità di mantenere i dati privati integrando protocolli crittografici e appoggiandosi a storage esterni, se richiesto

3.2 Testnet

Energy Web mette a disposizione una testnet che permette a chiunque sia interessato alle peculiarità della loro offerta di testare tutti i servizi senza spendere token reali.
La testnet si chiama Volta, e non è dissimile alle testnet che affiancano Ethereum.

Di seguito le differenze fra Volta e la EWC [20]:

	Volta	EWC
Data di lancio	Aprile 2019	Giugno 2019
Funzione primaria	Pre-produzione	Produzione
Token	90M + 10M di compensi transazioni e faucet	90M + 10M di compensi transazioni
Tariffe	Le risorse usate non hanno valore monetario	Valore monetario delle transazioni in base al gas usato
Caratteristiche	5 secondi per block limite di 8M gas	5 secondi per block limite di 8M gas
Connessione ad ETH	Bridge su Kovan Test Net	Bridge su Ethereum mainnet con un ERC-20 token
Nodi validatori	3 al lancio max 150	10 al lancio max 150

3.3 Algoritmo di consenso Proof of Authority

L'algoritmo che permette di assicurare un consenso sullo stato della blockchain fra tutti i nodi è PoA, più precisamente l'algoritmo Aura [21][22].
Nel caso di EWC, i nodi validatori sono gestiti dai partner dell'associazione. Si tratta, nella maggior parte dei casi, di aziende leader del settore energetico.
In breve, il funzionamento è il seguente:

1. Tutti i nodi validatori possiedono una lista aggiornata degli altri nodi validatori, oltre alla copia completa della blockchain e ad alcuni metadati ad essa collegati (es. il throughput della blockchain)

2. Per una finestra temporale ben definita, un validatore primario viene scelto dall'algoritmo e svolge il compito di raccogliere le transazioni e produrre il nuovo blocco. La scelta del validatore primario è in funzione del timestamp e del numero di validatori

3. Se il validatore primario non riesce a produrre il blocco (es. problemi hardware) o il blocco non viene convalidato dagli altri nodi (es. problemi di connessione), il prossimo validatore primario riprenderà dalle transazioni rimaste in sospeso

4. Gli altri validatori controllano che le transazioni siano legittime e firmano il blocco per poi propagarlo alla rete

5. Se la maggioranza dei validatori ha verificato il blocco, questo è aggiunto alla blockchain

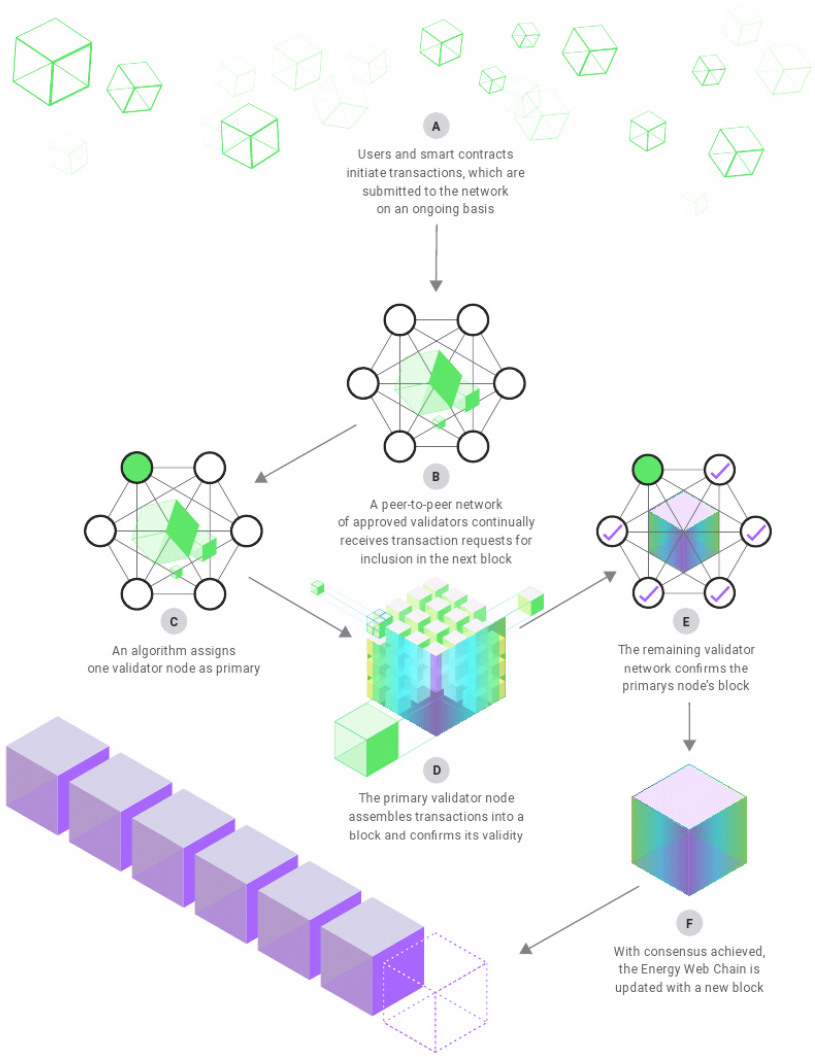


Figura 3.1: Passi dell'algoritmo PoA [23]

3.4 Costi di transazione

Una transazione è una qualsiasi operazione che modifica lo stato della blockchain. Trasferire token fra account, creare un nuovo smart contract o modificarne lo stato sono esempi di transazioni.
Ogni transazione ha un costo che sarà pagato in EWT. Grazie alla specificità di utilizzo della EWC e all'algoritmo di consenso, che insieme contribuiscono a ridurre la congestione della rete aumentandone il throughput, le tariffe sono generalmente molto basse e vanno da 0.00001 a 000000.1 EWT [24].
Il costo monetario da pagare per poter effettuare una transazione lo si può calcolare a partire dai seguenti fattori:

- **Il costo in gas:** il gas rappresenta la complessità computazionale necessaria per risolvere la transazione. Più è complessa l'operazione più gas sarà richiesto
- **Prezzo del gas:** valore dell'unità di gas in EWT. Prima di effettuare una transazione, l'utente stabilisce quanto è disposto a pagare per unità di gas. Più è alta l'offerta, maggiore sarà la priorità della transazione
- **Valore di mercato del token:** se si vuole ottenere il costo della transazioni in moneta fiat, basta effettuare la conversione fra EWT spesi e il loro valore monetario

Riassumendo con una formula [25]:
$costo(\$) = costo\ gas(gas) * prezzo\ gas(token/gas) * valore\ token(\$/token)$.

3.5 Transazioni private

Una delle caratteristiche fondanti di ogni blockchain è la possibilità di tracciare e verificare ogni transazione che avviene sulla chain.
Questa proprietà fondamentale, però, non è sempre desiderabile: anche per ottemperare alle regolamentazioni vigenti riguardanti la privacy, è necessario che ci sia la possibilità di nascondere informazioni sensibili a soggetti non autorizzati.
Sono stati quindi individuati diversi approcci per risolvere il problema. Di seguito sono elencati i più promettenti e avanzati. Per una lista completa è possibile consultare la documentazione [26].

3.5.1 Parity Secret Store

Un'implementazione promettente per risolvere il problema è il Secret Store del client Parity Ethereum.
L'idea è quella di generare una coppia di chiavi, pubblica e privata. Quella pubblica è nota, ed è utilizzata per crittografare i messaggi. Tuttavia, per decriptare i messaggi, è necessaria la chiave privata. Questa viene distribuita fra n nodi, in modo che per poter accedere alle informazioni protette sia necessario che un numero arbitrario m su n di questi sia favorevole.
La tecnica crittografica utilizzata è la "elliptic curve threshold encryption" [27].
L'implementazione è quella di un Key Management System (KMS), che invece di essere centralizzato è distribuito. La decentralizzazione elimina il Single Point of Failure (SPO), in quanto nessun nodo singolo ha accesso all'intera chiave.
Il sistema di consenso può sfruttare un sistema di votazione simile a quello usato per la governance.
Vi sono, però, vari limiti all'uso di questa tecnica. Infatti, è richiesto una setup iniziale fra i nodi interessati, compresa una rete separata per i servizi e le API. Un numero di nodi attivi deve essere garantito in ogni momento per raggiungere il limite inferiore di votanti. Infine, attualmente l'implementazione è limitata al client Parity / OpenEthereum.

Parity Private Transactions

Continuando a costruire sopra il Parity Secret Store, si può immaginare di generare uno smart contract criptato (Fig. 3.2). Questo verrebbe poi inserito dentro uno smart contract tradizionale pubblico sulla blockchain.
A quel punto, solo gli account autorizzati, con le chiavi necessarie, sono in grado di leggere e modificare lo stato del contratto interno. La nuova transazione viene eseguita e firmata off-chain, e il nuovo stato, criptato, viene reinserito nella blockchain pubblica.

3.5.2 Precise Proofs

Precise Proofs è uno schema basato sugli alberi di Merkle, in grado di verificare l'autenticità della struttura dati rivelando solo un subset di dati.
Si parte realizzando un albero di Merkle a partire dai dati interessati e si rende pubblica la radice.
Un terzo ente, interessato a verificare la validità di un subset di dati, riceverà solo i dati effettivamente richiesti e le hash dei dati non necessari. In questo

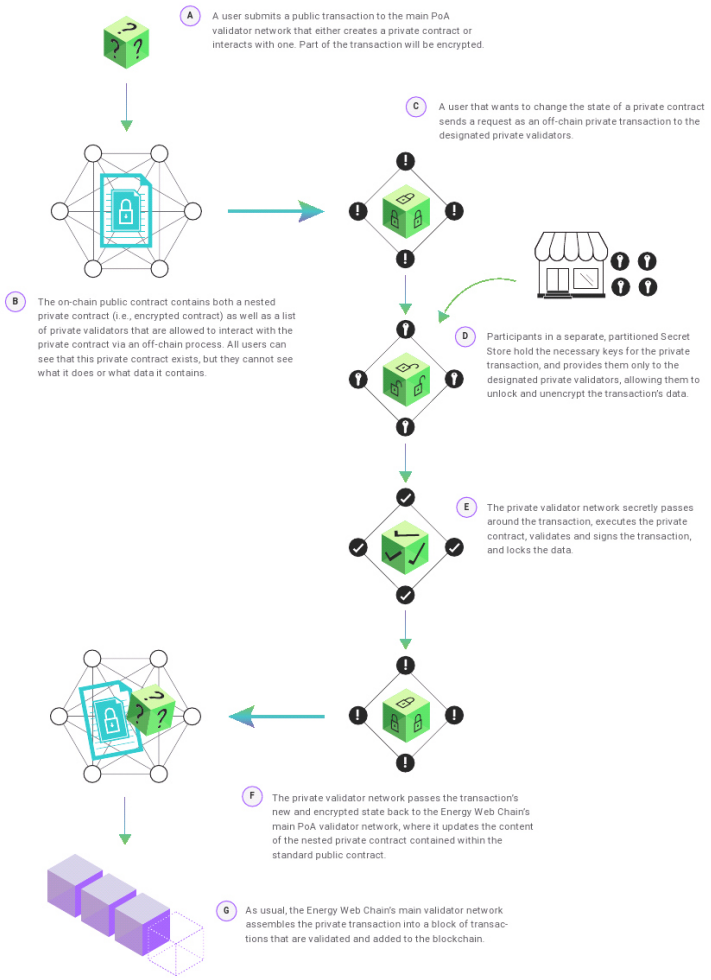


Figura 3.2: Utilizzo di Parity Private Transaction [28]

modo si hanno abbastanza informazioni da poter ricostruire la radice minimizzando i dati resi pubblici.

3.5.3 Zero Knowledge Proof Protocols

Gli Zero Knowledge Proof Protocols sono una categoria di protocolli con lo scopo di validare una computazione senza rivelarne alcun input (es. validare una transazione senza rivelare mittente, destinatario, movimento etc.).

zkSNARKs

zkSNARKs è probabilmente il protocollo più avanzato in questa categoria, sviluppato dal team Zcash [29]. Sebbene sia il più maturo, presenta un tallone d'Achille non indifferente.
Per poter utilizzare la Zero Knowledge Proof è necessario realizzare un circuito aritmetico che rappresenta la computazione da provare e generare da quello una coppia di chiavi prover/verifier. La creazione di questo setup deve essere sicura e pattuita a priori, perché se dovesse essere compromesso un avversario potrebbe approfittarne per realizzare prove false.

3.6 Storage

Il fatto che tutti i dati siano immagazzinati sulla blockchain per essere accessibili si traduce in un incremento della memoria necessaria per salvare l'intera blockchain su un dispositivo fisico, in una maggiore congestione della rete con conseguente calo di prestazione e in un aumento di costi possibilmente evitabili nelle esecuzione degli smart contract.
Una prima soluzione è di evitare di immettere dati sulla blockchain, e limitarsi ad utilizzare gli hash degli stessi, per poi limitarsi a verificarne la validità off-chain.
Nei casi in cui il salvataggio dei dati fosse necessario, si può provare ad optare per una delle tante soluzioni che offrono un servizio di storage distribuito.
Le principali soluzioni sono InterPlanetary File System (IPFS) e Storj. Per una lista completa è possibile consultare la documentazione [30]

InterPlanetary File System (IPFS)

IPFS [31] è un modello di filesystem distribuito. Invece di indirizzare i file con la loro locazione, li si identifica con un hash ottenuto dal loro contenuto. Se il file è troppo grande, lo si spezza in più parti con lo stesso risultato.
I nodi che fanno parte di questa rete si auto-iscrivono in delle tabelle di Distributed Hash Tables (DHT) che tengono traccia dei nodi che possiedono il file associato ad uno specifico hash.
Se si vuole contribuire alla rete, una volta scaricato il file lo si tiene in cache, fornendolo ad altri utenti che lo richiedano.
Il fatto che un file sia identificato dal suo hash garantisce inoltre l'integrità del dato e la sua immutabilità. Per aggiornare un file, infatti, bisogna utilizzare il sistema di versioning previsto dal protocollo.

Storj

Storj [32] è molto simile ad IPFS, con l'incentivo che i nodi sono pagati per svolgere la loro funzione di content-storage. Ovviamente il contratto prevede che l'host sia in grado fornire su richiesta in qualsiasi momento tutti i file che afferma di possedere.

3.7 Governance

La natura decentralizzata della blockchain rende un qualsiasi cambiamento che riguarda la sua infrastruttura un problema non banale.
Per gestire al meglio i possibili aggiornamenti che la EWC potrebbe necessitare, si è deciso di affidarsi a chi la comprende bene, cié gli sviluppatori. Il diritto di voto che determina quali proposte di modifica vengano accolte e quali respinte sarà determinato dalla quantità di gas che un singolo sviluppatore riesce a "generare" con i propri smart contracts. L'idea è che il valore che lo sviluppatore fornisce all'intero sistema rappresenta il suo peso nella votazione [33].
Ciò non esclude che, in caso si renda necessario, alcune scelte troppo complesse possano essere prese da persone designate senza passare per il meccanismo di voto della blockchain.

3.8 Staking

Per fornire quello che nell'architettura di ES-DOS è chiamato "Utility layer", è necessario il contributo di operatori che mettano a disposizione le proprie risorse a chi ha intenzione di sviluppare un servizio. Perché il tutto sia affidabile e adatto ad un ambiente di produzione, devono essere garantiti dei livelli di servizio, in maniera analoga ad un qualsiasi Service-Level Agreement (SLA) proposto da un fornitore di SaaS.
La soluzione proposta non è dissimile a quella implementata in altre blockchain, e si rifà al modello Proof of Stake (PoS). Gli attori in questo modello vengono suddivisi in due categorie:

- **Service providers:** sono organizzazioni che mettono a disposizione i nodi di utility. Per essere approvato, un service provider deve poter dimostrare la propria identità e depositare in garanzia una quantità di EWT per un periodo multi-annale. Dopo essere stati approvati, i service providers possono aggiungere ulteriori nodi di utility incrementando proporzionalmente il deposito di EWT. Finché lo SLA continua ad

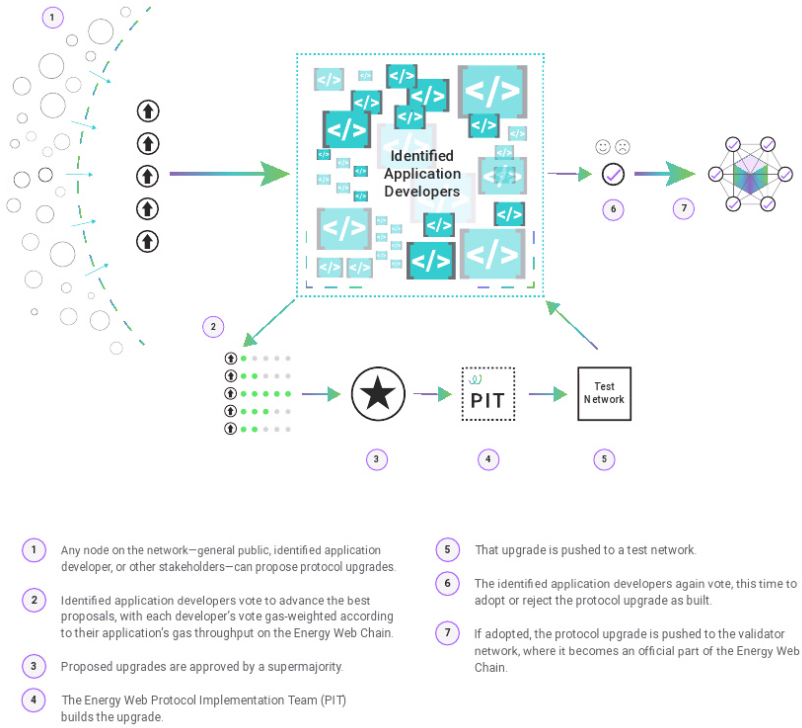


Figura 3.3: Passi per l'approvazione di una modifica all'infrastruttura di EWC [34]

essere rispettato, i service providers guadagneranno un interesse in base al loro deposito. In questo momento, il deposito minimo necessario per essere approvati come service provider si attesta fra i 10000 e i 100000 EWT, a cui si aggiungono fra i 1000 e i 10000 EWT per service node [35].

- **Patrons:** sono gli individui o le organizzazioni che finanziano un service provider depositando dei loro EWT per il service provider. Differentemente rispetto a ciò che accade per i service provider, non c'è un minimo alla somma depositata e questa può essere ritirata in qualsiasi momento. Questo porta i patrons a favorire service provider che rispettano gli SLA e che offrano il miglior modello di revenue-sharing.

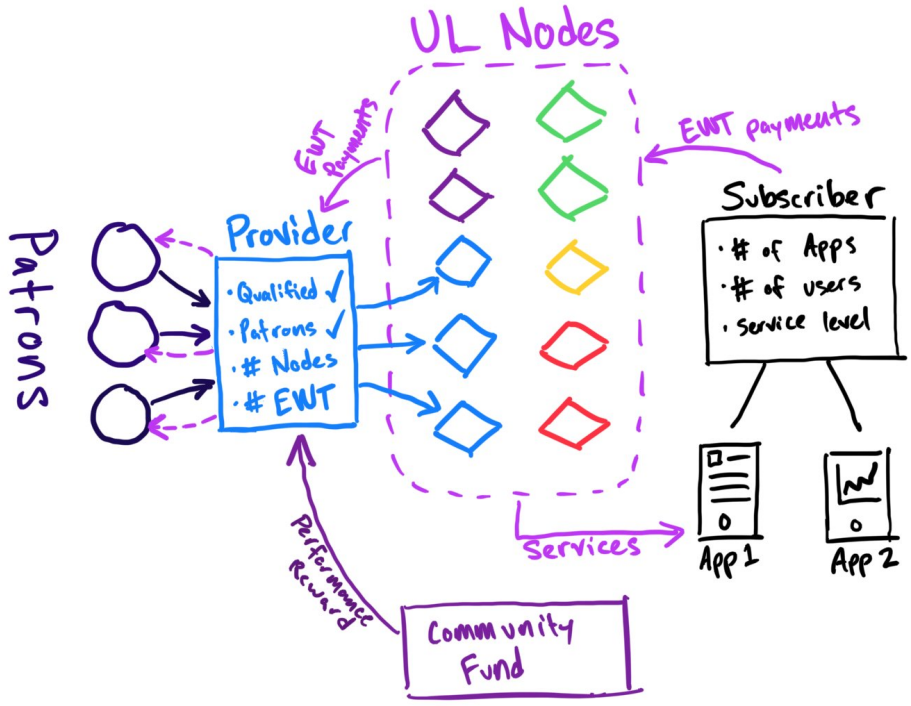


Figura 3.4: Come l'utility layer supportato dallo staking si integra in EW-DOS [35]

Capitolo 4

Servizi di Energy Web

4.1 Energy Web Name Service (EWNS)

L'Ethereum Name Service (ENS) è un sistema di denominazione distribuito, aperto ed estensibile basato sulla blockchain di Ethereum, ed una sua implementazione è stata resa disponibile anche sulla EWC.
Analogamente a un DNS, gli indirizzi vengono mappati su nomi di dominio facilmente memorizzabili, dando all'utente un'alternativa all'utilizzo degli indirizzi esadecimali.
L'implementazione disponibile pubblicamente consente prevede: indirizzi, indirizzi inversi, hash di dati, definizioni ABI, chiavi pubbliche, interfacce di smart contract e altro [36].

Per un utente finale il tutto è completamente trasparente. È compito dello sviluppatore di DApp integrare le funzionalità dell'ENS.
Esistono diverse librerie che già supportano questa funzione, anche se potrebbe essere necessaria qualche patch manuale.
Il procedimento per risolvere un dominio con ENS, comunque, non è particolarmente complicato:

1. Normalizzare ed applicare una funzione di hashing sul nome [37].

2. Chiamare la funzione resolver() sul registro ENS, passando l'output del passaggio 1. Ciò restituisce l'indirizzo del resolver responsabile del dominio

3. Usando l'interfaccia del resolver, chiamare la funzione addr() sull'indirizzo del resolver restituito nel passaggio 2, passando come parametro l'hash trovato nel passaggio 1. In alternativa, se non si sta cercando un indirizzo, chiamare la funzione dell'interfaccia prevista, ad esempio text()

4.2 Decentralized IDentifiers (DIDs)

I DID sono identificatori univoci usati per realizzare identità digitali verificabili e decentralizzate. Sono la base per l'implementazione delle Self Sovereign Identity (SSI) [38].
Il soggetto identificato può essere qualsiasi cosa: una persona, un'organizzazione o un bene fisico, per fare qualche esempio.
Un DID si comporta come un URI e punta ad un DID document, archiviato su un archivio pubblico permanente, come una blockchain. Questo documento contiene tutte le caratteristiche (claims), ad esempio certificazioni o autorizzazioni, che si riferiscono al corrispettivo DID.
Chiunque può creare un nuovo DID in qualsiasi momento.

Figura 4.1: Composizione di un DID [39]

Un DID è una stringa composta da tre parti (Fig. 4.1): lo schema did:, un metodo e un identificatore univoco rispetto al metodo [40].
I documenti DID, invece, sono dei file JSON-LD [41]. Il documento DID contiene tutte le informazioni, o claims, relative al soggetto identificato dal DID e fornisce una serie di meccanismi che consentono a un controller del DID di dimostrare il proprio controllo sul DID. Tipicamente esprimono anche metodi di verifica, come chiavi pubbliche crittografiche, e servizi relativi alle interazioni con il soggetto DID.
Il controller di un DID è l'entità che ha la capacità di apportare modifiche a un documento DID. Questa capacità in genere è subordinata dal possesso dal possesso della coppia di chiavi crittografiche. Si noti che un DID potrebbe avere più di un controller e il soggetto DID può essere il controller DID o uno di essi.
I sistemi che supportano la registrazione dei DID e la restituzione dei dati necessari per produrre documenti DID sono chiamati "verifiable data registry".
Un risolutore DID è il componente che, dato un DID, restituisce il documento corrispondente. Questo processo è chiamato risoluzione DID (Fig. 4.2).

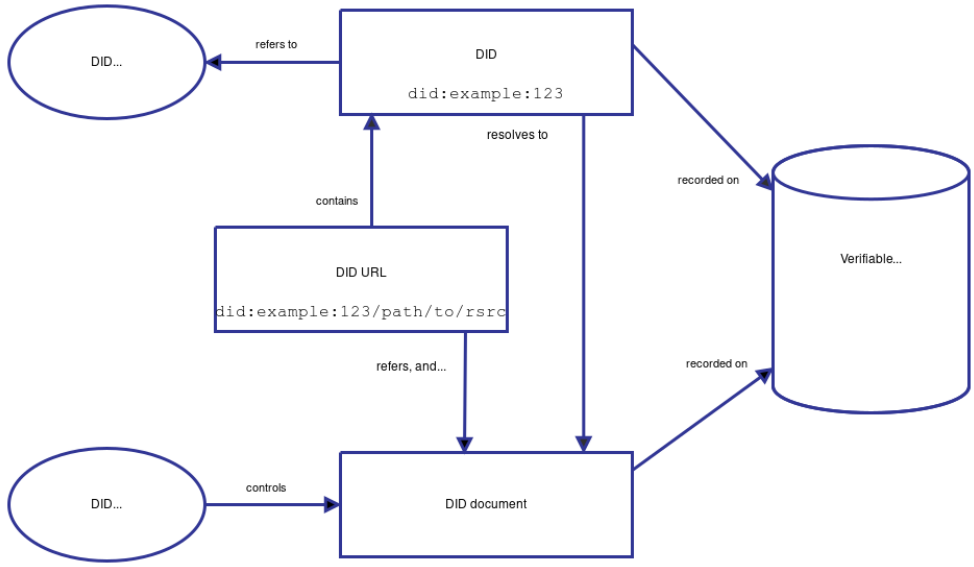


Figura 4.2: Panoramica dell'architettura dei DID e le relazioni fra le componenti fondamentali [42]

Il soggetto identificato ha bisogno di un'identità per essere in grado di dimostrare di possedere determinati tratti o caratteristiche a un verificatore. Il verificatore ha un certo numero di emittenti di fiducia ed è disposto ad accettarne i claims, cioè le affermazioni. Il soggetto può chiedere a uno di questi emittenti di aggiungere una dichiarazione firmata sul proprio documento DID. Inoltre, sia l'emittente che il verificatore devono essere in grado di verificare che il soggetto sia effettivamente il proprietario dell'identità. Ciò si ottiene tramite DIDAuth.
Questo processo può assumere molte forme, come specificato nel documento DID, ma un modo comune è attraverso un challenge inviata dall'emittente o dal verificatore al soggetto, del quale conoscono la chiave pubblica grazie al DID document, che a sua volta risponderà firmando con la propria chiave privata, dimostrando la propria identità.

Analogia con il mondo reale

Si vuole comprare alcool al bar. Per farlo, si deve essere in grado di dimostrare al barista di avere almeno 18 anni. Il barista in questo caso è il verificatore. Si può dimostrare la richiesta mostrando la propria carta d'identità. Questa viene rilasciato dal governo, che funge da emittente di fiducia per il barman, il quale verificherà che il richiedente sia effettivamente la persona raffigurata nella foto e si convincerà che del possesso del requisito della maggiore età.

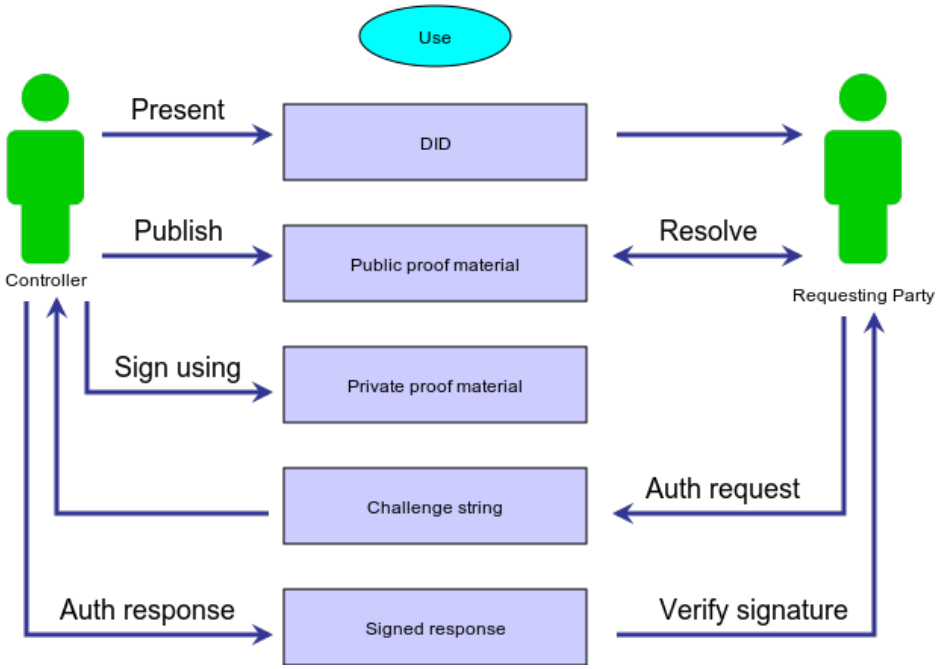


Figura 4.3: Panoramica dell'architettura dei DID e le relazioni fra le componenti fondamentali [43]

Le blockchain, grazie alle loro caratteristiche, si sono rivelate un sistema ideale per integrare un sistema di SSI, anche in ambiti che esulano il settore energetico [44].
Nel progetto di EW i DID ricoprono un ruolo centrale. Ogni utente può richiedere un DID che mantenga la lista di claims verificabili che l'utente possiede. È anche possibile assegnare un DID ad ogni DER, così da tenere traccia di tutte le informazioni e certificazioni legate a quello specifico DER.

Altri vantaggi forniti da questo sistema sono:

- Meccanismo di autenticazione - Il soggetto è in grado di fornire una prova crittografica del proprio controllo sul DID
- Autorizzazione e deleghe - Il protocollo prevede la possibilità per un DID di autorizzare o delegare un altro DID per permettergli di effettuare operazioni a suo nome

4.3 Identity and Access Management (IAM)

Identity and Access Management (IAM) è un insieme di policy in grado di garantire che solo le entità con le credenziali e le autorizzazioni necessarie possano accedere alle risorse.
Il sistema deve prima controllare l'identità digitale del soggetto e verificare i ruoli ad esso associati. L'operazione viene autorizzata solo se il soggetto ha i ruoli necessari. Per assicurare l'*accountability*, viene utilizzato un sistema di *logging* per registrare ogni operazione eseguita dal soggetto.

Per implementare un sistema IAM decentralizzato bisogna garantire alcune proprietà.

L'IAM deve essere resistente alla censura. Significa che non esiste alcun attore nel sistema che abbia il potere di limitare selettivamente le informazioni a cui gli altri hanno accesso. Questo aspetto è importante perché informazioni riguardanti situazioni casi in cui un'autorizzazione o una chiave è stata revocata o invalidata o se è stata appena concessa un'autorizzazione devono essere disponibili pubblicamente.
Una blockchain, per sua natura, soddisfa questi requisiti. Inoltre, le prove crittografiche contenute nelle transazioni consentono la verifica off-line dei dati. Le descrizioni dei ruoli sono contenute in uno smart contract ENS, mentre l'avvenuta concessione di un ruolo e i suoi dettagli è annotata nel documento DID dell'utente, mantenuto off-chain.

Le informazioni su utenti e ruoli devono essere riservate. Nella soluzione di Energy Web, nessuna terza parte dispone dell'elenco completo degli utenti di un'applicazione.
Ciò è garantito dal processo di concessione dell'autorizzazione, poiché è l'utente che archivia il claim e ne aggiunge la descrizione sulla blockchain, consentendogli di dimostrare che gli è stato concessa una determinata autorizzazione Fig. 4.4. Poiché si tratta di una operazione svolta dall'utente stesso, i contenuti del ruolo non sono divulgati [45]. L'unica informazione che un osservatore può raccogliere è che l'utente ha aggiunto un claim al proprio DID document, ma non il contenuto, l'origine o la natura del claim.

Anche l'IAM rappresenta un punto cardine nel progetto EW.
Attraverso il suo utilizzo, è possibile creare delle organizzazioni con una struttura gerarchica, simile ai domini attualmente utilizzati sul web. I ruoli che gli utenti possiedono determinano le azioni che possono compiere all'interno dell'organizzazione.
La gestione di tutti questi aspetti è stata resa molto più accessibile grazie alla DApp Switchboard (`https://switchboard.energyweb.org/`).

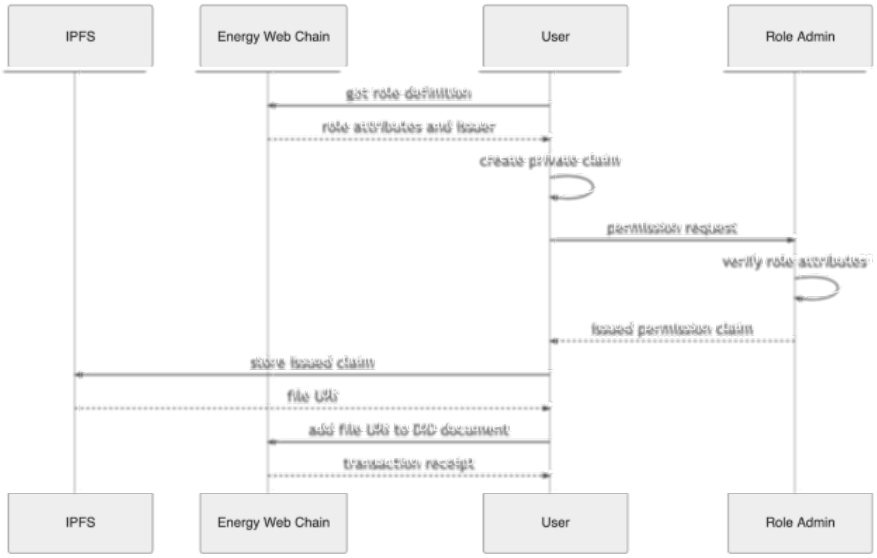


Figura 4.4: Processo di rilascio dei ruoli IAM [45]

Conclusione

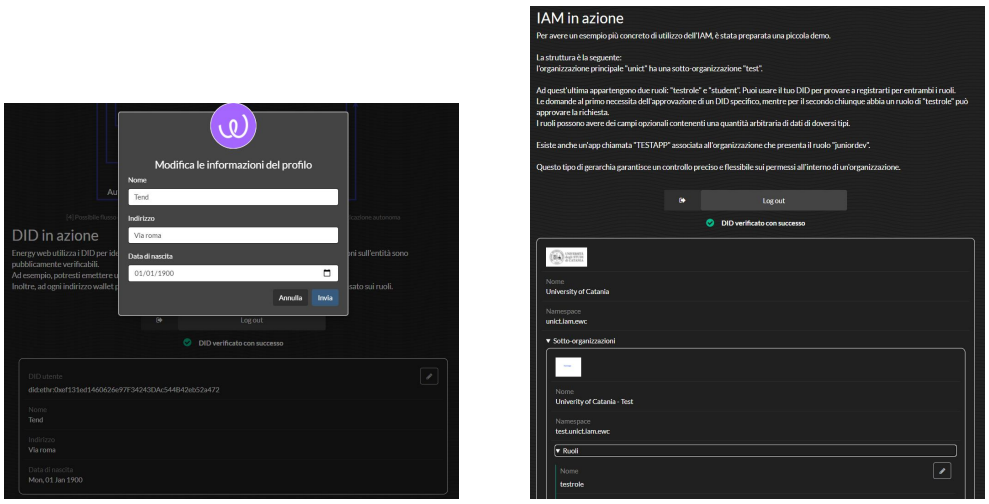
È ragionevole assumere che la decentralizzazione della rete elettrica sia un processo inevitabile che coinvolgerà un numero sempre crescente di utenze, sia nel ruolo di consumatori che nel ruolo di produttori.

Fra le possibili soluzioni, quella di usare la blockchain come infrastruttura in grado di supportare questo mercato sempre più decentralizzato sembra essere particolarmente promettente.
Ad una analisi approfondita EW risulta essere un progetto con una visione ed un focus chiari, guidato da persone che hanno le conoscenze necessarie per creare una piattaforma ad hoc per questo settore.
Proprio in virtù di ciò, le tecnologie che EW impiega non sono necessariamente innovative. Si cerca, al contrario, di seguire ed implementare quanti più standard possibile, e partire da questi per fornire dei servizi che rendano appetibile l'intera infrastruttura.
La scelta di rendere tutto il materiale, le implementazioni e le librerie open-source è in linea con la volontà di fornire degli strumenti che facilitino la vita agli sviluppatori, sui quali poi ricade il compito di creare DApp e servizi per l'utente finale.
È bene, infine, tenere a mente che EW, al momento della stesura di questo documento, è un progetto in divenire, e, per questo, soggetto a continui cambiamenti. Nonostante ciò, penso che EW fornisca quantomeno un ottimo caso di studio che valga la pena approfondire per chi fosse interessato ad affrontare la problematiche trattate nell'introduzione del documento.

Progetto

Ad affiancare questa relazione è stata realizzata una demo implementativa che è possibile provare navigando all'indirizzo `https://tendto.github.io/EW-showcase/`. Al fine di visionare tutte le funzioni, è necessario dotarsi dell'estensione browser MetaMask, connettersi alla rete Volta e avere a disposizione qualche Volta token.

La DApp, che rappresenta la sintesi del progetto, ha come scopo quello di mostrare in azione le funzionalità EWNS (sez. 4.1), DID (sez. 4.2) e IAM (sez. 4.3), utilizzando, ove possibile, le librerie e i frameworks messi a disposizione da EW.



(a) DID **(b)** IAM

Figura 4.5: Screenshots della DApp

L'intera documentazione e il codice dell'implementazione sono disponibili nella repository pubblica EW showcase (`https://github.com/TendTo/EW-showcase`) su Github.

Acronimi

DApp Decentralized Application. 6, 9, 25, 27, 28

DER Distributed Energy Resource. 4, 5, 9, 10, 24

DHT Distributed Hash Tables. 17

DID Decentralized IDentifiers. 8, 9, 22, 23, 24, 28

EAC Energy Attribute Certificate. 9

ENS Ethereum Name Service. 21

ERC Ethereum Request for Comment. 7

EVM Ethereum Virtual Machine. 7

EW Energy Web. 6, 24, 25, 27, 28

EW-DOS Energy Web Decentralized Operating System. 6, 7, 9

EWC Energy Web Chain. 6, 7, 8, 11, 12, 14, 18, 19, 21

EWNS Energy Web Name Service. 8, 11, 28

EWT Energy Web Token. 7, 14

IAM Identity and Access Management. 25, 26, 28

IoT Internet of Things. 10, 11

IPFS InterPlanetary File System. 17

KMS Key Management System. 15

P2P Peer to Peer. 5

PoA Proof of Authority. 7, 11, 12, 13

PoS Proof of Stake. 18

RES Renewable Energy Sources. 4, 9

SLA Service-Level Agreement. 18, 19

SPO Single Point of Failure. 15

SSI Self Sovereign Identity. 22, 24

TSO Transmission System Operator. 4

Bibliografia

[1] Terna S.P.A. *Il ruolo di Terna*. URL: https://www.terna.it/it/sistema-elettrico/ruolo-terna.

[2] Jūratė Jaraitė e Andrius Kažukauskas. «The profitability of electricity generating firms and policies promoting renewable energy». In: *Energy Economics* 40 (2013), p. 8. ISSN: 0140-9883. DOI: https://doi.org/10.1016/j.eneco.2013.10.001. URL: https://www.sciencedirect.com/science/article/pii/S0140988313002260.

[3] Aleksandar M. Mitrašinović. «Photovoltaics advancements for transition from renewable to clean energy». In: *Energy* 237 (2021), p. 5. ISSN: 0360-5442. DOI: https://doi.org/10.1016/j.energy.2021.121510. URL: https://www.sciencedirect.com/science/article/pii/S0360544221017588.

[4] ENEL x. *Gestione della generazione di energia: la chiave per la transizione*. 2020. URL: https://corporate.enelx.com/it/stories/2020/12/distributed-energy-resource-management.

[5] Lukas Stockburger et al. «Blockchain-Enabled Decentralized Identity Management: The Case of Self-Sovereign Identity in Public Transportation». In: *Blockchain: Research and Applications* (2021), p. 43. ISSN: 2096-7209. DOI: https://doi.org/10.1016/j.bcra.2021.100014. URL: https://www.sciencedirect.com/science/article/pii/S2096720921000099.

[6] Anselma Wörner et al. *Trading solar energy within the neighborhood: field implementation of a blockchain-based electricity market*. 2019. URL: https://energyinformatics.springeropen.com/articles/10.1186/s42162-019-0092-0.

[7] Shadab Alam et al. «Blockchain-based Initiatives: Current state and challenges». In: *Computer Networks* 198 (2021), p. 20. ISSN: 1389-1286. DOI: https://doi.org/10.1016/j.comnet.2021.108395. URL: https://www.sciencedirect.com/science/article/pii/S138912862100373X.

[8] Mohd Javaid et al. «Blockchain technology applications for Industry 4.0: A literature-based review». In: *Blockchain: Research and Applications* (2021), p. 30. ISSN: 2096-7209. DOI: https://doi.org/10.1016/j.bcra.2021.100027. URL: https://www.sciencedirect.com/science/article/pii/S2096720921000221.

[9] minimalsm. *Introduction to smart contracts*. 2021. URL: https://ethereum.org/en/developers/docs/smart-contracts/.

[10] Energy Web. *Energy Web History*. 2020. URL: https://www.energyweb.org/about/history/.

[11] Energy Web. *Energy Web Our Ecosystem*. 2020. URL: https://www.energyweb.org/work-with-us/our-affiliate-ecosystem/.

[12] Energy Web. *Energy Web What We Do*. 2020. URL: https://www.energyweb.org/about/what-we-do/.

[13] Energy Web. *EW-DOS: The Energy Web Decentralized Operating System.* An Open-Source Technology Stack to Accelerate the Energy Transition. PART 2: Technology Detail. 2020. URL: https://energyinformatics.springeropen.com/articles/10.1186/s42162-019-0092-0.

[14] Energy Web. «EW-DOS: The Energy Web Decentralized Operating System.» In: (2020). An Open-Source Technology Stack to Accelerate the Energy Transition. PART 2: Technology Detail, p. 14. DOI: https://www.energyweb.org/reports/EWDOS-Technology-Detail.

[15] Adam Z. Nagy. «Oracles with Chainlink on Ethereum networks tutorial series». In: *Medium* (2019). URL: https://medium.com/{@}aznagy/oracles-with-chainlink-on-ethereum-networks-tutorial-series-338b8a5f1726.

[16] Adam Nagy. *Data for your contracts: Oracles with Oraclize*. 2018. URL: https://energyweb.atlassian.net/wiki/spaces/EWF/pages/558432257/Data+for+your+contracts+Oracles+with+Oraclize.

[17] Jonathan von Waldenfels. *Understanding and using IPFS*. 2018. URL: https://energyweb.atlassian.net/wiki/spaces/EWF/pages/541687828/Understanding+and+using+IPFS.

[18] Wendell Cathcart. *Istanbul Hardfork*. 2020. URL: https://energyweb.atlassian.net/wiki/spaces/EWF/pages/916291657/Istanbul+Hardfork.

[19] Kiran Roy. *Chainspec update for Berlin Hardfork*. 2021. URL: https://energyweb.atlassian.net/wiki/spaces/EWF/pages/2568257551/Chainspec+update+for+Berlin+Hardfork.

[20] Wendell Cathcart. *Two Networks Serving Distinct Purposes*. 2020. URL: https://energyweb.atlassian.net/wiki/spaces/EWF/pages/717783156/Two+Networks+Serving+Distinct+Purposes.

[21] OpenEthereum. *Aura - Authority Round - Wiki*. 2021. URL: https://openethereum.github.io/Aura.

[22] Stephen Arsenault. *POA Network Whitepaper*. 2018. URL: https://github.com/poanetwork/wiki/wiki/POA-Network-Whitepaper.

[23] energyweb. *The Energy Web Chain*. 2018. URL: https://github.com/energywebfoundation/paper/blob/master/images/PoA_mechanism.jpg.

[24] Sam Hartnett. *How to Manage Transaction Costs on Public Blockchains*. 2019. URL: https://energyweb.org/2019/04/01/how-to-manage-transaction-costs-on-public-blockchains/.

[25] Sam Hartnett. *Transaction Costs Overview*. 2019. URL: https://energyweb.atlassian.net/wiki/spaces/EWF/pages/719847470/Transaction+Costs+Overview.

[26] Adam Nagy. *Privacy solutions overview*. 2018. URL: https://energyweb.atlassian.net/wiki/spaces/EWF/pages/610992129/Privacy+solutions+overview.

[27] Caimu Tang. «ECDKG: A Distributed Key Generation Protocol Based on Elliptic Curve Discrete Logarithm». In: (). DOI: http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.124.4128.

[28] energyweb. *The Energy Web Chain*. 2018. URL: https://github.com/energywebfoundation/paper/blob/master/images/private_transactions.jpg.

[29] Christian Reitwiessner. *zkSNARKs in a Nutshell*. 2016. URL: https://blog.ethereum.org/2016/12/05/zksnarks-in-a-nutshell/.

[30] Jonas Bentke. *On-Chain vs Off-Chain*. 2017. URL: https://energyweb.atlassian.net/wiki/spaces/EWF/pages/17760291/On-Chain+vs+Off-Chain.

[31] *IPFS*. URL: https://ipfs.io/.

[32] *STORJ*. URL: https://storj.io/.

[33] energyweb. *The Energy Web Chain*. 2018. URL: https://github.com/energywebfoundation/paper/blob/master/README.md#our-governance.

[34] energyweb. *The Energy Web Chain*. 2018. URL: https://github.com/energywebfoundation/paper/blob/master/images/protocol_upgrade_process.jpg.

[35] Sam Hartnett. «The Energy Transition is at Stake». In: (2021). How the EWT Escrow Model Unlocks the Full Potential of the Energy Web Tech Stack. DOI: http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.124.4128.

[36] Adam Nagy. *Using ENS*. 2020. URL: https://energyweb.atlassian.net/wiki/spaces/EWF/pages/555745281/Using+ENS.

[37] Nick Johnson. *Name Processing*. 2021. URL: https://docs.ens.domains/contract-api-reference/name-processing.

[38] Alexander Mühle et al. «A survey on essential components of a self-sovereign identity». In: *Computer Science Review* 30 (2018), p. 7. ISSN: 1574-0137. DOI: https://doi.org/10.1016/j.cosrev.2018.10.002. URL: https://www.sciencedirect.com/science/article/pii/S1574013718301217.

[39] *Decentralized Identifiers (DIDs) v1.0*. 2021. URL: https://www.w3.org/TR/did-core/#a-simple-example.

[40] *Decentralized Identifiers (DIDs) v1.0*. 2021. URL: https://www.w3.org/TR/did-core/.

[41] *JSON-LD material*. URL: https://json-ld.org/learn.html.

[42] *Decentralized Identifiers (DIDs) v1.0*. 2021. URL: https://www.w3.org/TR/did-core/#architecture-overview.

[43] *Decentralized Identifiers (DIDs) v1.0*. 2021. URL: https://www.w3.org/TR/did-use-cases/#use.

[44] Lukas Stockburger et al. «Blockchain-Enabled Decentralized Identity Management: The Case of Self-Sovereign Identity in Public Transportation». In: *Blockchain: Research and Applications* (2021), p. 43. ISSN: 2096-7209. DOI: https://doi.org/10.1016/j.bcra.2021.100014. URL: https://www.sciencedirect.com/science/article/pii/S2096720921000099.

[45] Micha Roon. *Digging Deeper into Self-sovereign Identity and Access Management*. 2020. URL: https://medium.com/energy-web-insights/digging-deeper-into-self-sovereign-identity-and-access-management-e6eefbac631e.