Contents

Abstract

Breve introduzione alla Energy Web Chain (EWC), la blockchain utilizzata da Energy Web

1 Differenze con Ethereum

Non è un mistero che la EWC sia molto ispirata ad Ethereum. Alcuni dei servizi offerti, come l'Energy Web Name Service (EWNS) sono fork con modifiche minime di progetti nati su Ethereum.
Tuttavia, ci sono alcuni aspetti e funzionalità che hanno necessitato una revisione più profonda, per evitare non ostacolassero la visione dietro il progetto.

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, se richiesto

2 Testnet

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

Di seguito il confronto fra Volta e la EWC

	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	Kovan Test Net	ERC-20 token
Nodi validatori	3 al lancio max 150	10 al lancio max 150

3 Algoritmo di consenso POA

L'algoritmo che permette di assicurare un consenso sullo stato della blockchain fra tutti i nodi è POA, più precisamente l'algoritmo Aura [8][1].
Nel caso di EWC, i validatori sono i partner dell'associazione. Si tratta, nella maggior parte dei casi, di aziende leader del settore energetico.
In breve, il funzionamento è il seguente:

- 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)
- 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 block. La scelta del validatore primario è in funzione del timestamp e del numero di validatori
- Se il validatore primario non riesce a produrre il block (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
- Gli altri validatori controllano che le transazioni siano legittime e firmano il block per poi propagarlo alla rete
- Se la maggior parte dei validatori hanno verificato il blocco, questo è aggiunto alla blockchain

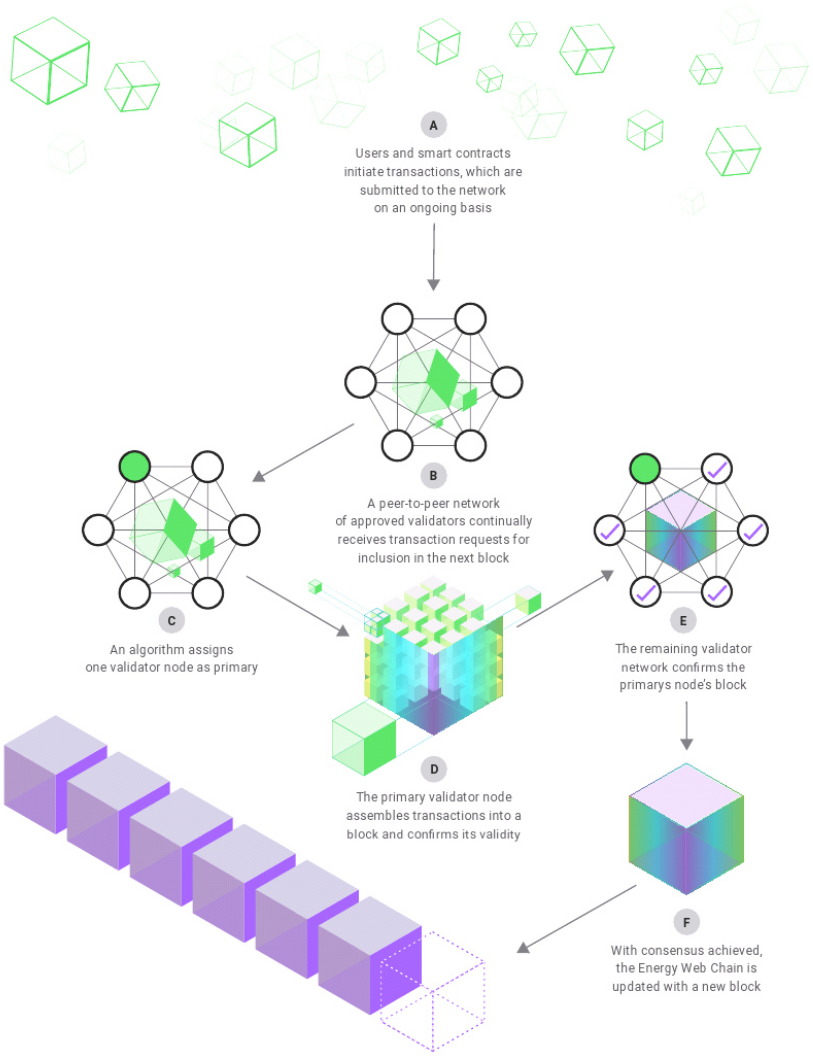

Figure 1: Passi dell'algoritmo POA [3]

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.
Il costo da pagare per poter effettuare una transazione dipende 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:
$costo(\$) = costo\ gas(gas) * prezzo\ gas(token/gas) * valore\ token(\$/token)$.

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 da occhi indiscreti.
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 [7]

5.1 Parity Secret Store

Un'implementazione promettente per risolvere il problema è il Secret Store di 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" [9].
L'implementazione è quella di un Key Management Sysyem (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.

5.1.1 Parity Private Transactions

Continuando a costruire sopra il Parity Secret Store, si può immaginare di generare uno smart contract criptato. 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.

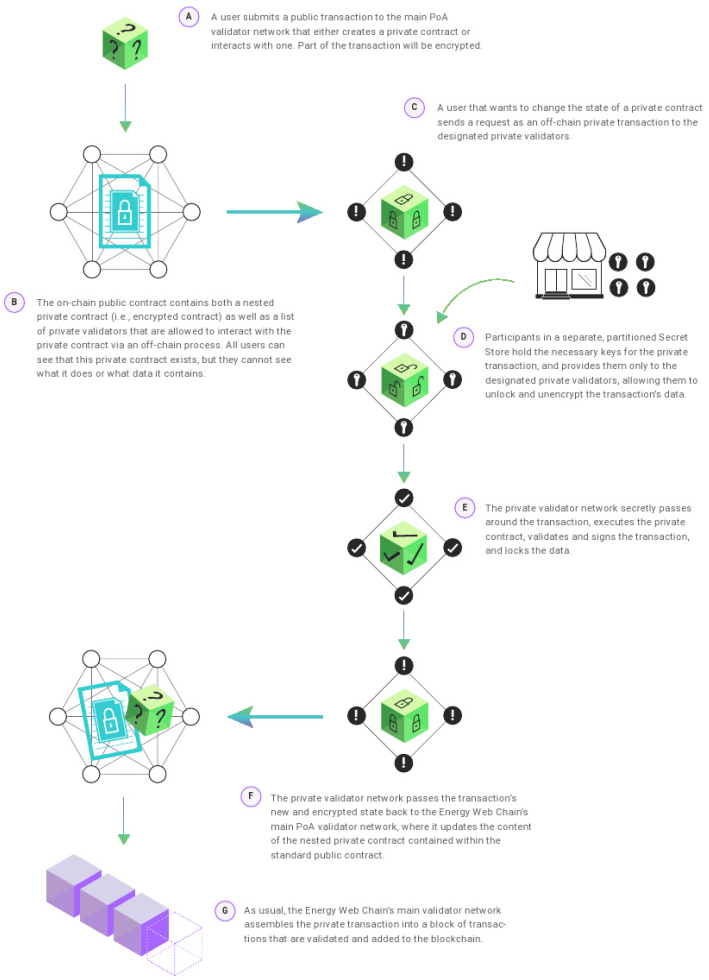

Figure 2: Utilizzo di Parity Private Transaction [4]

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

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.).

5.3.1 zkSNARKs

zkSNARKs è probabilmente il protocollo più avanzato in questa categoria, sviluppato dal team Zcash. 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, perchè se dovesse essere compromesso un avversario potrebbe approfittarne per realizzare prove false.

6 Storage

Il fatto che tutti i dati siano immagazzinati sulla blockchain per essere accessibili si traduce in un incremento spesso più che lineare della memoria necessaria per salvare l'intera blockchain su un dispositivo fisico, cosa decisamente non auspicabile se si vuole favorire la partecipazione anche di piccoli dispositivi (IoT).
Una prima soluzione è di evitare di immettere dati sulla blockchain, e limitarsi ad utilizzare gli hash degli stessi, per poi poter verificarne la validità off-chain.
Nei casi in cui questo non fosse possibile, si può provare ad optare per una delle tante soluzioni che offrono un servizio di storage distribuito. Le principali soluzioni sono IPFS Storj. Per una lista completa è possibile consultare la documentazione [2]

6.0.1 IPFS

IPFS 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, fornendo 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 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.

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, cioè 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 contracs. L'idea è che il valore che lo sviluppatore fornisce all'intero sistema rappresenta il suo peso nella votazione. [5]
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.

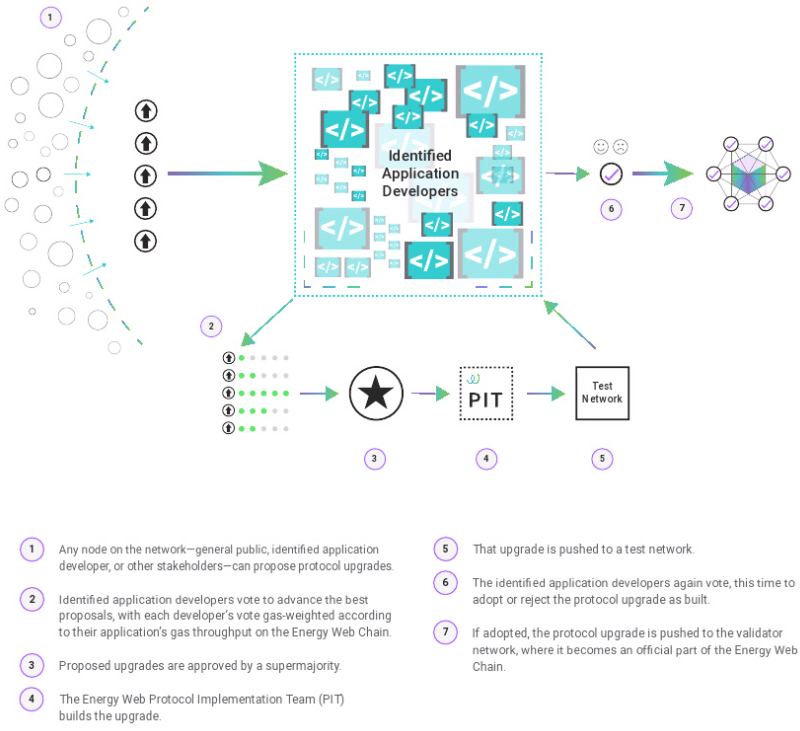

Figure 3: Passi per l'approvazione di una modifica all'infrastruttura di EWC [6]

References

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

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

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

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

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

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

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

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

[9] 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.