Alma Mater Studiorum · Università di Bologna

SCUOLA DI SCIENZE
Corso di Laurea in Informatica per il Management

La Blockchain e lo sviluppo della moneta di Tezos

Relatore: Chiar.mo
Prof. SANGIORGI DAVIDE

Presentata da:
VATALARO GIADA

Sessione
Anno Accademico
2018/2019

INDICE

INTRODUZIONE

La moneta di Tezos è l'argomento centrale del presente studio, ma prima di arrivare a parlare di essa faremo un'infarinatura sulla *Blockchain*, la quale viene sempre di più legata al tema delle *criptovalute* che a loro volta sono collegate agli *smart contracts*. La **tecnologia Blockchain** ha ricevuto negli ultimi anni un'attenzione quasi senza precedenti: questo documento è un'immersione negli aspetti tecnici che hanno interessato la Blockchain, infatti nel primo capitolo andremo a comprenderne **l'evoluzione nella storia**, lo **sviluppo delle basi** e la **crittografia** andando a studiare la sicurezza che garantisce. Nel secondo capitolo andremo a sviluppare il concetto delle **criptovalute**, novità molto importante negli ultimi decenni che sta portando dei cambiamenti radicali per quanto riguarda l'evoluzione del sistema di pagamento, difatti le valute virtuali hanno catturato sempre un maggiore interesse, sia per l'introduzione di tecniche innovative per quanto riguarda le operazioni di pagamento e di trasmissione di moneta che per la rivoluzione non solo in ambito economico ma anche in ambito sociale. **Bitcoin** è stata la primissima criptovaluta che nel 2009 ha utilizzato la blockchain, una rete di scambi peer to peer in cui non vi è autorità centrale incaricata di registrare le transazioni. Poi verrà fatta una panoramica sulla criptovaluta **Ethereum** creata nel 2013 da Vitalik Buterin e definita come il "*world computer*" (computer del mondo), infrastruttura decentralizzata che esegue *smart contracts*; per concludere parleremo della **moneta di Tezos**, sviluppata da Arthur Breitman, che fondamentalmente è il fulcro del documento, la quale è una piattaforma basata su *smart contracts* e avente un *cripto-registro generico* e *auto-*

correttivo. Nel terzo ed ultimo capitolo, parleremo delle parti più salienti della moneta di Tezos, partendo dall'algoritmo di "**Proof of Stake**", utilizzato per raggiungere il consenso in un sistema a catena di blocchi; verrà anche fatta un'introduzione sul meccanismo di **auto-modifica** che caratterizza questa moneta; parleremo degli smart contracts e del linguaggio di programmazione utilizzato**, Michelson** e **Liquidity**, descrivendo anche il **ciclo di vita che compie un'operazione** nel momento in cui entra in un blocco fino a quando non arriva nella blockchain di Tezos, e infine parleremo della **sicurezza riguardante questi contratti intelligenti**, i quali non sono propriamente legali nel mondo in cui viviamo oggi, infatti ancora non vengono utilizzati come i contratti cartacei ma sono validi solo su Internet. Nel corso del documento si potrà visualizzare l'evoluzione di tutti questi argomenti, valutando le basi che hanno portato a quello che sono oggi.

Capitolo 1

1.1 Blockchain: l'evoluzione nella storia

La Blockchain è una tecnologia sviluppata nel 2008 con il rilascio di un paper intitolato "Bitcoin: A Peer-to-Peer Electronic Cash System"[1] e pubblicato sotto lo pseudonimo di Satoshi Nakamoto (autore la cui identità è tuttora sconosciuta). La pubblicazione di questo paper di sole nove pagine ha posto le basi per il sistema di pagamento *trustless*[2] basato su tecnologia blockchain, unendo una serie di meccanismi già noti ma trovando soluzioni innovative al pagamento distribuito tra persone distanti, senza la necessità di un ente centrale che garantisca la validità dei pagamenti. In realtà, ben dieci anni prima di questo paper da parte di Satoshi Nakamoto, viene fondata l'idea di uno strumento di pagamento virtuale in cui però era ancora necessario un ente centrale per garantire le transazioni. In seguito, l'idea di assicurare l'anonimato delle transazioni all'interno delle reti telematiche deriva dal contesto culturale in cui si sviluppa il movimento *cypherpunks*[3]*,* ossia da un gruppo di soggetti (John Gilmore, Eric Hughes, Tim May) che crearono una mailing list dove poter discutere i temi della privacy e della cifratura dei dati. Lo scopo principale era quello di contrastare le possibili restrizioni della libertà e del diritto alla privacy, derivanti dalla diffusione delle tecnologie informatiche, le quali avrebbero dato la possibilità alle

[1] Paper di Satoshi Nakamoto: https://bitcoin.org/bitcoin.pdf

[2] L'essere *Trustless* è una proprietà specifica di Bitcoin che permette a due o più persone di potersi scambiare valore senza il bisogno di conoscersi e quindi di fidarsi l'uno dell'altro e senza la possibilità di imbrogliare, il tutto senza il controllo di un'autorità centrale, ma solo grazie a regole matematiche e algoritmi di crittografia.

[3] I *Cypherpunks* usano la crittografia per proteggere la privacy intesa come "diritto dell'individuo di rivelarsi in modo selettivo al mondo".

grandi società di monitorare e controllare le informazioni sugli individui potendo invadere i loro stili di vita. Nel 1993 viene pubblicato il "Cypherpunk Manifesto"[4] e nel 1997 si stava cercando di risolvere il problema dello *spam* nella posta elettronica facendo in modo che l'invio dei messaggi non desiderati fosse considerato un gesto oneroso, mentre nel 1998 viene pubblicato il primo sistema decentralizzato di pagamento garantito dalla cifratura e dalla "*proof of stake*", un tipo di protocollo per la messa in sicurezza di una rete di criptovaluta e per il conseguimento di un consenso distribuito; basato sul principio che ad ogni utente venga richiesto di dimostrare il possesso di un certo ammontare di criptovaluta; si differenzia dai sistemi *proof-of-work*[5] che sono basati su algoritmi di *hash* che validano le transazioni elettroniche, in maniera da incentivare i partecipanti ad agire in modo onesto sul network potendo altrimenti perdere i fondi depositati in caso di validazione di transazioni fraudolenti. Negli stessi anni, Nick Szabo, un informatico statunitense, con studi legali e di crittografia, laureatosi presso l'Università di Washington nel 1989 in informatica, propone la definizione di "*smart contract" (*contratti intelligenti); Szabo scrive: "L'idea di base dello smart contract è che molti tipi di clausole contrattuali (come la garanzia, l'assunzione dell'obbligazione, la delimitazione di un diritto di proprietà, ecc.) possono essere incorporati nell'hardware e nel software che trattiamo, in modo da rendere la violazione del contratto costosa (se desiderato, addirittura

[4]Cypherpunk Manifesto: https://cypherpunkholdings.com/wp-content/uploads/A-Cypherpunks-Manifesto.pdf.

[5] Il concetto originale di Proof-of-Work risale al 1993, anno in cui è stato sviluppato per prevenire attacchi denial of service e altre violazioni come network spam. Sostanzialmente questa soluzione richiedeva un po' di lavoro all'utente del servizio, solitamente inteso come tempo di lavorazione di un computer.

proibitiva) per il soggetto inadempiente"; inoltre, gli smart contracts sono capaci di eseguire in modo autonomo delle transazioni. Nel 2004, Hal Finney[6], basandosi sui principi di *HashCash*, moneta digitale prima di Bitcoin, basata su un primo tentativo della proof of work, si trattava di una valuta digitale, che per un certo tempo ebbe anche successo, l'HashCash si poneva come obiettivo anche la capacità di resistere ad una serie di accatti hacker fra cui quelli DDoS[7] (Distributed DoS). L'uso di una forma di Proof of Work (PoW) doveva anche limitarne la distribuzione ed aveva molte possibilità potenziali poi sfruttate dal *Bitcoin* e da altre *criptovalute*. Ciò che portò alla decadenza HashCash fu il superamento dell'algoritmo Proof of Work da parte della capacità di elaborazione dei processori, che ne portò all'iperinflazione, successivamente nel 2005 Nick Szabo pubblica una proposta avente ad oggetto il *Bitgold,* una valuta digitale decentralizzata in cui per la prima volta si poneva l'accento sulla distribuzione comunitaria della gestione della valuta sempre utilizzando l'algoritmo *Proof of Work* per limitarne la produzione ed evitare fenomeni inflazionistici, tutto ciò mettendo alla base l'idea che aveva sviluppato Hal Finney. Quindi, in sintesi, sono queste le basi che portano nel 2008 alla pubblicazione del paper di Satoshi Nakamoto in cui viene descritto il funzionamento di Bitcoin, che poi porta, il 3 gennaio del 2009, alla creazione del "*Genesis Block*" ossia del blocco iniziale della Blockchain Bitcoin. Per la creazione di questi sistemi di pagamento viene utilizzata una moneta elettronica anonima ed altri

[6] Il geniale Hal Finney, crittografo scomparso nel 2014, fu uno dei primi a lavorare con Satoshi e partecipò alla prima transazione bitcoin di sempre.

[7] Attacchi DDoS sono utilizzati per distrarre l'attenzione da altre attività criminali simultanee, ad esempio truffe bancarie, oppure contro istituzioni governative o finanziare, come quelli rivendicati da Anonymous, o anche contro siti di e-commerce per motivi di concorrenza.

strumenti di pagamento non tracciabili, il tutto utilizzando tecnologie crittografiche su larga scala, permettendo anche di creare sistemi di messaggistica sicuri, contratti digitali e sistemi di identità digitale senza invadere la privacy. La proposta di Nakamoto mette insieme alcune delle tecnologie note a quei tempi, come la cifratura a chiavi asimmetrica, utilizzata già a partire dal 2008, la quale permette di assicurare la paternità di un messaggio e la sua integrità attraverso il diverso utilizzo di una chiave pubblica ed una chiave privata di cifratura. In secondo luogo, la blockchain viene progettata da Nakamoto distribuendola su un network peer-to-peer[8] in cui i singoli computer degli utenti operano come "*peer*" o "*nodi*" agendo da distributori e fruitori delle informazioni. Infine, riemerge il principio della "proof-of-work" in cui l'idea principale è la competizione della potenza di calcolo, dove il nodo che esegue il meccanismo di consenso (chiamato minatore) utilizza la sua risorsa di calcolo per l'operazione di *hashing* per poter vincere il diritto di generare il nuovo blocco con bonus. Con "hashing" si intende il processo che genera un output di dimensione fissa partendo da un input di dimensioni variabili. Questo viene fatto attraverso l'uso di formule matematiche conosciute come funzione di hash (implementate come algoritmi di hashing). Anche se non tutte le funzioni di hash implicano l'uso di crittografia, le cosiddette funzioni crittografiche di hash sono alla base delle criptovalute e grazie ad esse la blockchain e altri sistemi distribuiti sono in grado di raggiungere livelli significativi di integrità e sicurezza dei dati. In generale possiamo definire la proof-of-work (PoW) sia

[8] Il termine peer-to-peer si riferisce allo scambio di criptovalute o asset digitali attraverso un network distribuito. Quindi una piattaforma P2P consente ad acquirenti e venditori di eseguire operazioni senza la necessità di intermediari.

come meccanismo di creazione del consenso per la validazione delle transazioni, sia come strumento di incentivazione per i partecipanti a mettere a disposizione risorse computazionali. Detto ciò, la blockchain viene definita come il primo passo verso l'evoluzione.

1.2. Caratteristiche e funzionamento della tecnologia Blockchain

La blockchain è spesso definita come un database distribuito, pubblico e crittografico, ma in realtà è molto più di questo. Alla base della tecnologia vi è un "libro mastro distribuito" (distributed ledger): esso contiene tutte le informazioni relative ad un bene, inoltre consente l'eliminazione della terza parte. La tecnologia blockchain ha delle caratteristiche peculiari rispetto alle altre ad oggi maggiormente utilizzate. La gestione delle informazioni avviene attraverso una rete *peer-to-peer*, queste funzionalità rendono la blockchain una tecnologia trasparente, sicura e decentralizzata e con una capacità di archiviazione pressoché illimitata. Grazie al meccanismo *peer-to-peer* ciascuno svolge un ruolo contemporaneamente attivo per la creazione e la validazione delle transazioni e passivo per la conservazione della memoria delle stesse. Il database delle informazioni è distribuito su tutti i computer del network: se un Paese dovesse decidere di bloccare l'accesso al network tutte le transazioni sarebbero conservate in ciascuno dei nodi, potendole ripristinare in qualsiasi momento. A

rendere ancora più sicura questa tecnologia sono i nodi, che essendo anonimi permettono di confermare le transazioni in completo anonimato. Blockchain significa letteralmente "catena di blocchi", quindi ogni blocco viene concatenato al precedente e al successivo utilizzando la crittografia per creare dei blocchi crittografati. Possiamo, perciò, affermare che la blockchain è una "Internet delle transazioni" ed è considerata la soluzione di molti problemi legati alla sicurezza e alla proprietà dei dati sensibili. Un'altra particolarità di cui dispone questa nuova tecnologia è l'irreversibilità delle transazioni effettuate, infatti è un registro immodificabile, nel senso che le informazioni non sono conservate da un'unica persona, ma sono presenti sui computer di tutti i partecipanti. I dati registrati sono ripudiabili da coloro che li hanno generati e possono essere sempre verificati. A tal proposito, il sistema conserva i metadati e le informazioni di contesto delle singole transazioni, rendendo riconducibili le stesse agli account partecipanti al network. Altra caratteristica peculiare della blockchain è il meccanismo di incentivazione dei partecipanti, infatti la tecnologia spinge gli utenti ad agire in buona fede, incentivando la partecipazione al sistema, tramite meccanismi di guadagno e rendendo estremamente difficili le condotte abusive. Un elemento centrale in ogni blockchain è l'algoritmo di *mining*[9], infatti i miner sono i primi ad essere interessati quando bisogna effettuare una transazione. Ad esempio: ci sono due persone, Alice e Bob, che usano Bitcoin. Alice deve due bitcoin a Bob ma per inviarglieli deve prima

[9] Il mining (o nodi validatori) è un processo basato sulla pura statistica dove ogni tentativo di hashing ha la stessa probabilità di essere quello buono. Inoltre, le transazioni tra utenti vengono verificate e aggiunte al registro pubblico della blockchain e viene utilizzato per introdurre nuove monete nella fornitura circolante.

trasmettere un messaggio con la transazione che vuole eseguire a tutti i miner del network. Così facendo fornisce l'indirizzo pubblico di Bob, la somma di Bitcoin che vuole inviare e una firma digitale, tutto questo insieme alla sua chiave pubblica. La firma viene generata con la chiave privata di Alice e i miner possono verificare che Alice è effettivamente in possesso dei Bitcoin e che vuole eseguire la transazione. Nel momento in cui i miner sono sicuri che la transazione è valida possono inserirla in un blocco insieme a tante altre transazioni, per poi cercare di minare il blocco. L'output che deve ritornare dovrebbe iniziare con un determinato numero di 0 per poter essere considerato valido. Il numero di 0 necessari dipende da un fattore chiamato "difficoltà" di mining, che varia a seconda della potenza di calcolo presente all'interno del network. Per produrre un hash che inizi con il numero di 0 desiderato, i miner aggiungono un numero casuale chiamato "nonce"[10] all'interno del blocco prima di inserirlo nell'algoritmo. Trovato l'hash valido i miner trasmettono il nuovo blocco a tutti gli altri miner, i quali verificano che il blocco sia effettivamente valido per poter aggiungerlo alla propria copia della blockchain e completare la transazione. Inoltre, nel blocco bisogna anche aggiungere l'hash di output del blocco precedente per fare in modo che tutti i blocchi siano concatenati, da cui deriva il nome blockchain (catena di blocchi). Ciascun miner possiede la propria copia della blockchain nel proprio computer e tutti si fidano della blockchain più lunga e quindi di quella per cui è stato investito il maggior lavoro computazionale. Se un miner cambia una transazione in un blocco precedente, l'hash di output per quel blocco

[10] In crittografia il termine *nonce* indica un numero, generalmente casuale o pseudo-casuale, che ha un utilizzo unico; nonce deriva dall'espressione inglese "for the nonce", che significa appunto "per l'occasione".

cambierà, portando alla modifica di tutte le hash successive a causa del collegamento tra blocchi. Il modello utilizzato per produrre blocchi alla catena è chiamato Proof-of-Work (PoW), un algoritmo di consenso originale in una rete Blockchain che viene anche utilizzato per confermare le transazioni e quindi si riferisce al processo di lavoro dei Miners. Con questo meccanismo i minatori competono l'uno con l'altro per completare le transazioni sulla rete e vengono premiati. La procedura che seguono i miners è la seguente:

- Incrementano un numero arbitrario scelto casualmente, nell'intestazione del blocco, denominato *nonce*;
- Calcolano l'hash dalla nuova intestazione;
- Controllano se l'hash dell'intestazione (espresso numericamente) è inferiore ad un predeterminato valore obbiettivo (target).

Tuttavia, se la conversione numerica dell'hash non è inferiore al valore target, il blocco sarà espulso dal network. Infatti, l'obbiettivo di questa procedura (PoW) è di trovare un blocco che abbia un hash il cui valore sia il più basso possibile. Tutto ciò rende la blockchain una tecnologia protetta, infatti il protocollo di comunicazione blockchain alimenta le varie criptovalute tramite l'utilizzo di tecniche crittografiche con lo scopo di assicurare e verificare le transazioni. La tecnologia blockchain è stata impiegata da varie criptovalute presenti sul mercato, tra cui Bitcoin, Ethereum e Tezos, di cui parleremo nel prossimo capitolo.

1.3 Crittografia: la sicurezza della Blockchain

La tecnologia Blockchain dal punto di vista della sicurezza è una recente svolta nel mondo dell'informatica sicura senza autorità centralizzata di un sistema di rete aperto. I registri della blockchain archiviano e conservano dati in blocchi, i quali sono organizzati secondo una sequenza cronologica e sono connessi tra loro attraverso prove crittografiche. La creazione della tecnologia Blockchain ha portato diversi vantaggi in una vasta gamma di settori, garantendo una maggiore sicurezza in contesti *trustless*. Tuttavia, la sua natura decentralizzata comporta anche degli svantaggi: infatti, se confrontate con i tradizionali database centralizzati, le blockchain presentano un'efficienza limitata e richiedono una maggiore capacità di archiviazione. Per quanto riguarda i vantaggi possiamo affermare che i dati blockchain sono spesso archiviati in migliaia di dispositivi all'interno di un network distribuito di nodi, il sistema e i dati sono molto resistenti ad errori tecnici e attacchi malevoli. Ciascun nodo del network può archiviare e replicare una copia del database, quindi non esiste un unico punto di rottura: infatti, un singolo nodo che va offline non influisce sulla disponibilità o la sicurezza del network. Al contrario, diversi database convenzionali fanno affidamento su un unico o su pochi server e sono più vulnerabili a guasti e attacchi informatici. I blocchi registrati sulla blockchain sono estremamente difficili da rimuove o modificare: infatti è improbabile che vengano invertiti e annullati, quindi questa tecnologia è ideale per conservare registri finanziari o qualsiasi altro tipo di dati che necessitano di tracciabilità. In quasi tutti i sistemi di pagamento tradizionali, le transazioni non dipendono solo dalle due parti coinvolte ma anche da un intermediario (una banca, un

provider di servizi di pagamento o una società di carte di credito) mentre usando la tecnologia blockchain l'intermediario non è più necessario, in quanto il network distribuito di nodi verifica le transazioni attraverso un processo conosciuto come mining. Per questo motivo la Blockchain è spesso definita un sistema *trustless*. Altro fattore determinante del sistema blockchain è il fatto di negare il rischio legato alla necessità di fidarsi di una singola organizzazione e di ridurre le commissioni e i costi complessivi eliminando intermediari e terze parti. Allo stesso tempo, uno degli svantaggi legati a questa tecnologia è riferito alla stabilità, che se da un lato è considerata uno dei vantaggi della Blockchain dall'altro non lo è. La possibilità di non poter modificare più i dati una volta aggiunti alla blockchain è uno degli aspetti negativi che interessano questo argomento. Infatti, cambiare dati o codice blockchain è in genere molto complicato e spesso richiede un *hard fork,* cioè una modifica al protocollo di una criptovaluta che risulta incompatibile con le versioni precedenti, quindi i nodi non si aggiornano alla nuova versione e perciò non saranno in grado di elaborare transazioni o aggiungere nuovi blocchi alla blockchain. Gli hard fork possono essere usati per modificare o migliorare un protocollo esistente, oppure per creare un nuovo protocollo e una nuova blockchain indipendenti. Inoltre, la Blockchain è molto sicura grazie all'utilizzo della crittografia a chiave pubblica (o asimmetrica), infatti ciascun account blockchain (o indirizzo) ha due chiavi corrispondenti: una *chiave pubblica* (che può essere comunicata) e una *chiave privata* (che andrebbe tenuta segreta). Per decifrare un determinato messaggio gli utenti hanno bisogno di una delle due chiavi, questo perché la coppia di chiavi è legata

matematicamente da una funzione. Il funzionamento della crittografia asimmetrica è tanto semplice quanto efficace, quindi chiunque sia in possesso della chiave pubblica usata per criptare il messaggio non sarà più in grado di decifrarlo; l'unico modo per farlo è di possedere la chiave privata associata alla chiave pubblica utilizzata. Questo meccanismo, però, non è così efficiente se utilizzato al contrario, perché se si dovesse criptare un messaggio usando la chiave privata, chiunque fosse in possesso della chiave pubblica associata sarebbe in grado di decifrarlo. A tal proposito è fondamentale custodire le informazioni relative alla propria chiave privata perché altrimenti chiunque potrebbe inviare dei beni per conto di qualcun altro. Inoltre, se un utente perde la sua chiave privata, i fondi sono effettivamente persi, e non c'è nulla che possa fare. Per affrontare questo problema è bene utilizzare un *wallet* (portafoglio) esso è un modo sicuro di memorizzare la chiave privata e pubblica. Attraverso la chiave privata il portafoglio consente di eseguire transazioni di routine come l'invio e la ricezione di *coin* o controllare il saldo complessivo. È come un numero di conto a cui è collegata tutta l'attività blockchain dei partecipanti. I wallet possono essere semplici, come una chiave privata scritta su un pezzo di carta oppure possono essere sofisticati dispositivi di archiviazione che memorizzano chiavi private e si connettono a Internet quando l'utente desidera eseguire una transazione. Un altro aspetto negativo è *l'inefficienza*, infatti le blockchain, soprattutto quelle che usano la Proof of Work, sono altamente inefficienti, questo perché il *mining* è molto competitivo e ci può essere solo un vincitore ogni dieci minuti mentre il lavoro di tutti gli altri va sprecato. Nonostante gli aspetti negativi, la tecnologia Blockchain presenta

dei vantaggi unici e quindi è destinata a rimanere. La strada verso il totale utilizzo di questo metodo è ancora lunga, ma già molti settori stanno sperimentando vari usi dei sistemi Blockchain.

Capitolo 2

Le criptovalute

2.1 Una prima definizione

Una criptovaluta è uno "strumento digitale utilizzato per effettuare acquisti e vendite attraverso la crittografia, al fine di rendere sicure le transazioni, verificarle, controllare la creazione di nuova valuta; denaro, moneta virtuale"[11]. Con il termine criptovaluta (o in accezione anglosassone *cryptocurrency*) si identifica un nuovo strumento di transazione che opera attraverso valute digitali. Una criptovaluta è costituita da una funzione primaria di mezzo di scambio all'interno di un sistema economico *peer-to-peer* che fa uso della crittografia per verificare e proteggere le transazioni. La grande maggioranza delle criptovalute è decentralizzata, mentre i sistemi bancari sono centralizzati. Inoltre, le criptovalute sono distribuite su molti computer sparsi in tutto il mondo e conosciuti come *nodi:* infatti chiunque abbia a disposizione una connessione a internet, o anche solo l'accesso ad un segnale radio, può trasferire valori attraverso continenti con un semplice click. A differenza dei trasferimenti bancari intercontinentali, le transazioni di criptovaluta hanno costi molto bassi e sono irreversibili, mentre le operazioni *charge-back*[12] permesse dalle società di carte di credito non lo sono. L'emissione e la gestione delle criptovalute sono determinate

[11] Definizione presa dal dizionario Treccani: "http://www.treccani.it/vocabolario/criptovaluta_res-016bf79f-8997-11e8-a7cb-00271042e8d9_(Neologismi)".

[12] Charge-back (riaccredito) è una procedura con la quale vengono gestiti i movimenti relativi a contestazioni da parte dei titolari di strumenti di pagamento (principalmente, carte di credito).

dall'architettura del network che è basata su algoritmi programmati e prove crittografiche. In questo momento storico la finanza digitale sta ricoprendo un ruolo estremamente importante al punto che stanno venendo alla luce due scenari fondamentali: nel primo si assisterà ad un cambiamento talmente radicale da cui difficilmente si potrà far ritorno; o viceversa, come viene sostenuto da molte figure di spicco della finanza internazionale, tra cui Warren Buffet[13], si assisterà ad un disastro finanziario globale. La nuova generazione della *digital economy* (economia digitale) vede positivamente le criptovalute; esse vengono sostenute per i loro punti di forza, tra i quali l'indipendenza dalle banche centrali. Perciò essendo decentralizzate le transazioni possono avvenire direttamente tra gli utenti, senza dover fare affidamento su un intermediario. Tuttavia, molte criptovalute non essendo rilasciate da enti governativi di nessun tipo, sono teoricamente immuni a qualsiasi manovra. A seconda della struttura del network e della distribuzione dei nodi, alcune criptovalute possono essere considerate più centralizzate di altre. La componente primaria di gran parte delle criptovalute è la tecnologia di cui abbiamo già parlato, la Blockchain, una catena di blocchi concatenati protetti crittograficamente. La prima criptovaluta decentralizzata che ha utilizzato la blockchain è stata *Bitcoin,* e attualmente esistono più di mille criptovalute diverse, chiamate anche *altcoin* o *alternative coin*, monete alternative, tutte con proprietà e casi d'uso differenti.

[13] Tratta da: "https://www.corriere.it/economia/18_gennaio_10/buffett-le-criptovalute-faranno-brutta-fine-81090898-f622-11e7-9b06-fe054c3be5b2.shtml".

2.2 Le diverse criptovalute

2.2.1 Bitcoin

La prima criptovaluta decentralizzata, *Bitcoin*, è stata creata nel 2009 da uno sviluppatore anonimo sotto lo pseudonimo di Satoshi Nakamoto. L'idea centrale era di creare un sistema di pagamento elettronico indipendente e decentralizzato basato su prove matematiche, crittografiche e su protocolli "Proof-of-Work". La tecnologia alla base di Bitcoin è progettata per preservare l'integrità dei dati e delle transazioni. Ogni transazione è contrassegnata da una firma digitale e verificata con delle tecniche crittografiche, in modo da garantire che i fondi non vengano spesi più di una volta. Quindi ogni volta che viene effettuata una transazione, passa dallo stato di "non confermata" a quello di "confermata" e solo in quest'ultimo momento ogni nodo "generatore" raccoglie tutte le transazioni "non confermate" che conosce in un "blocco" candidato, cioè un file che contiene un *hash crittografico* del precedente blocco valido. Ogni blocco è dotato di un "*header*" utilizzato per organizzare il database distribuito, all'interno di questo *header* è contenuto *l'hash* di tutte le transazioni registrate nel blocco. La funzione *hash* consente di ridurre in maniera univoca un insieme di bit in una stringa alfanumerica, univocamente riconducibile al contenuto originario, fornendo una sorta di "impronta digitale". I vari nodi sono in competizione tra loro per la generazione di ogni hash di chiusura di ciascun blocco della catena ed il primo che riesce a risolvere tale algoritmo comunicherà la soluzione nel network e successivamente questa verrà verificata dagli altri nodi. Se tale soluzione è

corretta il blocco è aggiunto alla blockchain e quindi salvato su tutti i nodi partecipanti al network, quando il blocco contenente le transazioni raggiunge sei conferme, il client Bitcoin viene confermato, così facendo la transazione viene registrata per sempre nella blockchain. Dal momento che i blocchi futuri sono dipendenti dai blocchi precedenti è impossibile modificare o eliminare un blocco. L'intero processo prende il nome di "*mining*", un termine analogo al *gold mining* (estrazione di oro): infatti è paragonato a quanto accade nelle miniere di estrattori d'oro, in quanto i miner consentono "l'estrazione" di nuovi bitcoins. L'emissione di bitcoins è determinata dall'algoritmo distribuito, il quale definisce la difficoltà del processo di validazione e la quantità di bitcoins che viene rilasciata nel "*wallet*"[14] (portafoglio) dei miners ogni volta che viene validato un blocco. Il minatore aggiunge un nuovo blocco alla catena solo dopo aver constatato che aggiungendo nuove informazioni al blocco precedente viene restituito un determinato *codice hash*[15]. Per risolvere il rompicapo il miner deve trovare un numero (chiamato *nonce*) che se processato tramite la funzione di hash, insieme ad altri dati presenti nel blocco da validare, deve restituire un hash che inizia con un determinato numero di zeri: questo hash viene chiamato "hash della testa del blocco". Il primo miner che risolverà il rompicapo avrà una ricompensa in bitcoin e successivamente tutti gli altri miner dovranno confermare la soluzione trovata per validare il blocco: in caso affermativo, il miner che ha validato il blocco otterrà la sua ricompensa. Qualsiasi tentativo di modificare un blocco già registrato comporterebbe la "rottura della catena", in quanto porterebbe alla modifica dell'hash di tale

[14] *Wallet*: file generato dal client Bitcoin contenente una serie di coppie chiave pubblica-chiave privata.

[15] Hash: funzione crittografica unidirezionale.

blocco ed a quella di tutti i successivi. Per alterare la blockchain di Bitcoin è necessario disfare l'intera struttura registro per registro, cosa praticamente impossibile anche per i computer più potenti. Il processo è regolato da un "algoritmo di consenso" chiamato Proof of Work, in cui il livello di sicurezza si basa sul fatto che i dati siano distribuiti attraverso una miriade di nodi situati in ogni parte del mondo, ciò significa che se viene modificato un nodo, gli altri partecipanti al network sono in grado di riconoscere chi è il nodo corrotto, in quanto non corrisponde a nessun'altra delle copie.

2.2.2 Ethereum

Grazie al grande successo ottenuto da Bitcoin sono nate molte altre criptovalute, che si differenziano da Bitcoin per numero di monete erogate, algoritmi utilizzati e tempi di attesa tra un blocco e il successivo. A tal proposito un'altra criptovaluta, quasi paragonabile a Bitcoin e di cui parleremo è *Ethereum*, sviluppata nel 2013 da *Vitalik Buterin*[16]. Ethereum è una criptovaluta, definita come il "world computer" (computer del mondo), è un'infrastruttura decentralizzata che esegue programmi chiamati "smart contract", cioè applicazioni che vengono eseguite esattamente come programmato, senza rischio di interruzioni, censura, fronde o intervento di terzi; inoltre è distribuita, pubblica e open source. In Ethereum, come in altre blockchain, più nodi seguono un protocollo in cui le transazioni dai client vengono raggruppate in blocchi e i nodi usano un protocollo di consenso per concordare i blocchi successivi. Ogni blocco include un hash crittografico del suo predecessore, rendendo difficile la manomissione del libro mastro. Gli *smart contract* hanno un valore molto importante nell'ambito della Blockchain introdotta da Ethereum, infatti rappresentano una sfida concorrenziale. Ethereum costruisce una blockchain con un linguaggio di programmazione completo Turing che consente a chiunque di scrivere contratti intelligenti e applicazioni decentralizzate dove poter creare le proprie regole. Ogni utente ha la proprietà di uno o più account mittente e un account destinatario, collegati a uno o più indirizzi. Un account Ethereum contiene quattro campi:

[16] Vitalik Buterin è un programmatore e scrittore russo, fondatore di Ethereum.

- Il *nonce*, un contatore usato per assicurarsi che ogni transazione possa essere elaborata una sola volta;
- Il *saldo etere corrente del conto*;
- Il *codice contratto* dell'account, se presente;
- La *memoria dell'account* (vuota di default).

Blockchain utilizza due tipi di account:

- *conti di proprietà esterna*;
- *conti di contratti*.

I primi sono controllati da persone, pertanto, analogamente a Bitcoin, ogni persona ha la propria chiave privata, che viene usata al fine di effettuare transazioni nella blockchain di Ethereum. Al contrario, invece, gli account di contratti sono controllati da un codice di contratto intelligente, quindi tali conti sono una sorta di *cyber-entità* e con il proprio saldo, possono essere attivati attraverso alcune transazioni, proveniente da un conto esterno (o da altri contratti). Gli smart contracts sono eseguiti dai minatori o nodi che propongono ripetutamente nuovi blocchi da aggiungere alla blockchain. Alla creazione di un blocco, il minatore seleziona una sequenza di transazioni del client ed esegue i suoi codici di contratti intelligenti in sequenza, in modo da trasformare il vecchio stato del contratto in un nuovo stato. Successivamente, i contratti intelligenti di quel blocco vengono rieseguiti da validatori o nodi che ricostruiscono lo stato attuale della blockchain. Ogni minatore convalida i blocchi proposti da altri minatori e i blocchi più vecchi sono convalidati dai minatori appena uniti o dai client che interrogano lo stato del contratto.

I vantaggi associati ai contratti intelligenti sono molteplici:

- *Indipendenza*, nessun obbligo di intermediari;
- *Risparmio*, costi ridotti;
- *Sicurezza*, nessun attacco hacker perché sono protetti dalla crittografia;
- *Precisione*, nessun errore.

Gli script o gli smart contracts su Ethereum vengono stipulati usando un nuovo linguaggio di programmazione creato appositamente per questa criptovaluta, chiamato *Solidity*. Inoltre, per incentivare gli utenti a mantenere i nodi ed eseguire gli script, Ethereum ha la propria criptovaluta, *ether* (ETH), in questo modo limita anche la spam all'interno del network. Per eseguire un'operazione su Ethereum bisogna pagare una tassa di esecuzione, chiamata "*Gas*". Il Gas misura la quantità di lavoro che un'azione necessita per la sua esecuzione. Tuttavia, i nodi danno maggiore priorità alle richieste che offrono un maggior profitto, quindi maggiore è il numero di computazione richiesto dall'operazione, maggiore sarà la quantità di Gas necessario. Ethereum, inoltre, offre diversi vantaggi, uno di questi riguarda i nodi: infatti se uno di loro dovesse smettere di funzionare ce ne sono tanti altri in tutto il mondo che mantengono online il servizio. Lo stesso vale per la censura, è molto più facile bloccare un server centralizzato rispetto a centinaia o migliaia in tutto il mondo. Questa funzionalità protegge il servizio garantendo che sia sempre disponibile a tutti, in qualsiasi parte del mondo. Allo stesso tempo, tra gli svantaggi del protocollo Ethereum troviamo i contratti di un blocco, i quali vengono eseguiti uno alla volta, quindi i minatori e i validatori non possono sfruttare le moderne architetture

multicore. Infatti, i contratti non possono essere eseguiti contemporaneamente in quanto si potrebbero avere dei conflitti di dati, ovvero accessi simultanei alle stesse variabili di archiviazione. Usando questa criptovaluta, è possibile effettuare pagamenti su altri account o sui computer che eseguono alcune operazioni richieste. Ecco perché Ethereum è definito "*denaro programmabile*".

2.2.3 Tezos

Tezos è la terza ed ultima criptovaluta di cui ci occuperemo, la quale è basata su Blockchain ed è una piattaforma di *smart contracts* per la creazione di applicazioni decentralizzate che non possono essere chiuse da terze parti. Inoltre, è un *cripto-registro generico* e *auto-correttivo*. Tezos è in grado di istanziare qualsiasi libro mastro basato su Blockchain. L'unicità di Tezos è data da:

- *Autoemendamento e aggiornabilità*: infatti Tezos può aggiornarsi tramite un processo di modifica nel protocollo senza la necessità di un "hard fork"[17].
- *Proof-of-Stake*, in cui il *baking* (cottura) è per Tezos ciò che il mining è per Bitcoin, i nodi in Tezos forniscono le risorse computazionali necessarie per mantenere la rete.

[17] Un hard fork è una modifica al protocollo di una criptovaluta che risulta incompatibile con le versioni precedenti, quindi i nodi che non si aggiornano alla nuova versione non saranno in grado di elaborare transazioni o aggiungere nuovi blocchi alla blockchain. Tratto da: "https://www.binance.vision/it/blockchain/hard-forks-and-soft-forks".

- *Sicurezza del contratto intelligente e verifica formale*, Tezos ha utilizzato un linguaggio di contratto intelligente chiamato "*Michelson*".

Lo sviluppatore di questa nuova moneta è *Arthur Breitman*, ingegnere informatico e matematico con esperienza nel settore finanziario. Lui ha seguito da vicino l'ascesa storica di Bitcoin fin dall'inizio e ha notato la sua "incapacità di evolversi", così ha deciso di sviluppare una "versione aggiornata" e tutto ciò ha dato vita al progetto Tezos. Ethereum, invece, è una delle criptovalute più vicine a Tezos in quanto a somiglianze, infatti è la seconda criptovaluta in termini di capitalizzazione di mercato, anche se entrambe hanno una funzionalità di smart contract. Le differenze tra le due criptovalute si trovano:

- nella *Governance* in cui le modifiche apportate alla Blockchain sono prestabilite dalla fondazione Ethereum e dai suoi sviluppatori, infatti operano seguendo una tabella di marcia prestabilita e le parti interessate hanno un'influenza limitata sul cambiamento. Tuttavia, con aggiornamenti e modifiche all'interno della blockchain di Tezos, viene utilizzato il sistema di *governance on-chain*. Quindi significa che gli sviluppatori possono inviare suggerimenti per la modifica della rete Tezos che viene quindi votata e, se si ottengono i voti richiesti, la modifica verrà effettuata e gli sviluppatori riceveranno un incentivo;
- nel *meccanismo di consenso* dove Ethereum si basa sulla Proof of Work e presto passerà alla Proof of Stake, mentre Tezos opera nell'ambito di un meccanismo di Proof of Stake delegato. La proof of

work utilizzata da Ethereum richiede un elevato numero di consumo di energia e i minatori devono dimostrare di aver esteso le risorse necessarie nel calcolo di un livello inferiore di un determinato obbiettivo. Dall'altra parte, il meccanismo della proof of stake di Tezos consente ai titolari dei token di delegare gli altri a raggiungere un accordo sullo stato della rete per loro.

- *nel linguaggio di programmazione per gli smart contracts*, infatti Tezos si basa su Michelson, mentre Ethereum sul linguaggio di programmazione Virtual Machine. La differenza più notevole tra i due linguaggi è che Michelson è scritto in un formato di testo leggibile dall'uomo invece le operazioni Ethereum Virtual Machine (EVM) sono rappresentate come byte. La seconda grande differenza tra Michelson e EVM è che gli elementi di dati di Michelson sono digitati, in generale un tipo è un'informazione che limita le possibili cose che possono essere fatte con un dato valore di dati. Quindi i tipi consentono al programmatore di comunicare le proprie intenzioni in modo più dettagliato alla macchina e consente alla macchina di comunicare al programmatore quando l'esecuzione di discosta da tali intenzioni.

2.3 La moneta di Tezos (XTZ)

Tezos è una piattaforma open-source per risorse e applicazioni che possono evolversi aggiornandosi. Una singola moneta di Tezos viene definita "*tez*" e l'unità più piccola è solo un centesimo. Per definire la moneta di Tezos si utilizza il simbolo ꜩ (\ua729, "lettera latina tz"), quindi 1 cent = ꜩ0,01 = un centesimo di tez. L'unicità di questa criptovaluta è dettata dal fatto che si possa andare oltre l'innovazione, e l'aggiornamento avviene tramite un processo di modifica del protocollo senza la necessità di un *hard fork*, inoltre vengono coordinate le parti interessate all'interno di una rete per un lungo periodo di tempo. Le parti interessanti di Tezos regolano gli aggiornamenti del protocollo principale. Esattamente come le blockchain partono da un hash della genesi, Tezos inizia con un "*protocollo seed*". Questo protocollo può essere modificato per riflettere qualsiasi algoritmo basato su blockchain, il meccanismo avviene esponendo al protocollo due funzioni procedurali:

- *Set_test_protocol,* che sostituisce il protocollo usato in testnet con un nuovo protocollo;
- *Promotion_test_protocol,* che sostituisce il protocollo corrente con il protocollo attualmente in fase di test.

Queste funzioni trasformano un contesto modificando il protocollo associato, il quale può essere modificato per varie condizioni. Nella sua versione più semplice, un voto delle parti interessate provoca un cambio di protocollo e progressivamente possono essere votate regole più complicate. Questo è un controllo efficace e algoritmico della "*costituzionalità*". L'upgrade, per gli sviluppatori che si basano su Tezos, offre una forte

garanzia che il protocollo funzionerà senza intoppi nel futuro. La criptovaluta Tezos è stata costruita per resistere alla prova del tempo. Inoltre, l'utilizzo della Proof of Stake è in totale contrasto con Bitcoin, infatti in Tezos ci si aspetta che i partecipanti (ovvero i "nodi") a Tezos raggiungono il consenso sullo stato della Blockchain, mentre in Bitcoin il consenso si basa sulla prova di lavoro (ovvero il mining). La *Proof-of-Stake* è un meccanismo basato su *baking (cottura)* e prevede la delega, quindi consente a qualsiasi *stakeholder*[18] di partecipare al consenso senza rinunciare alla custodia dei propri token. A giugno del 2018, Tezos è stata lanciata come una delle principali reti Proof-of-Stake. Altra caratteristica peculiare di XTZ è il *linguaggio di Michelson*, utilizzato per gli *smart contracts*, progettati tenendo conto della sicurezza e della verifica formale. Esso consente agli sviluppatori di dimostrare matematicamente che il codice funziona correttamente, secondo le sue specifiche formali o determinate proprietà. Tutto ciò si adatta perfettamente ai contratti smart finanziari, che rappresentano un valore reale significativo e richiedono garanzie che i fondi non vegano persi o congelati a causa di bus nel codice. Perciò, a differenza di altri linguaggi usati per le diverse criptovalute, *Michelson* può essere considerato molto sicuro.

[18] Stakeholder: Tutti i soggetti, individui od organizzazioni, attivamente coinvolti in un'iniziativa economica (progetto, azienda), il cui interesse è negativamente o positivamente influenzato dal risultato dell'esecuzione, o dall'andamento, dell'iniziativa e la cui azione o reazione a sua volta influenza le fasi o il completamento di un progetto o il destino di un'organizzazione. Tratta da: http://www.treccani.it/enciclopedia/stakeholder/

Capitolo 3

La Blockchain di Tezos

3.1 La Proof of Stake

La Proof-of-stake si riferisce ad una categoria di algoritmi che vengono utilizzati per raggiungere il consenso in un sistema a catena di blocchi. Quindi in particolare questo sistema previene gli attacchi di Sybil (cioè impedisce ad un singolo partecipante di mascherarsi da N altri partecipanti). In un sistema, il voto di un determinato partecipante è collegato direttamente al numero dii monete che ha in possesso, perciò se una persona ha 100 monete non può fingere di essere 1000 persone diverse con 100 monete ciascuna. Per riuscire ad avere una catena di blocchi che con il tempo progredisca è necessario creare nuovi blocchi e aggiungerli alla catena. Questo compito viene svolto dai minatori che competono per questo diritto spendendo la potenza di calcolo per risolvere enigmi crittografici casuali. Il vincitore arriva a creare il blocco successivo e si guadagna una ricompensa per averlo fatto. In questo paradigma, più potenza di calcolo ha un minatore, più è probabile che il blocco successivo venga creato. Al contrario, i sistemi Proof of stake ruotano intorno all'idea che più monete ha un minatore/validatore/produttore di blocchi, più è probabile che venga creato il blocco successivo. In generale abbiamo due classi di algoritmi di proof-of-stake:

- *Prova di prelievo a catena*, in cui per Bitcoin un validatore viene selezionato casualmente in ogni slot temporale per creare un blocco che si basa sulla catena più lunga. Tuttavia, invece di selezionare un validatore in base a chi risolve prima i puzzle crittografici, la probabilità di selezione è ponderata in base a quante monete si bloccano.
- *Tollerante agli errori bizantini (BFT) prova di prelievo,* in questo caso invece di un validatore casuale che ottiene il diritto di creare un blocco che ogni altro partecipante deve accettare, i sistemi BFT introducono l'idea di proporre e accettare. Come il sistema PoS a catena, un validatore selezionato casualmente viene scelto per proporre un blocco agli altri validatori e tutti i validatori onesti devono comunicare tra loro finché non si accordano. Una volta che si sono messi d'accordo, accettano il blocco che poi viene finalizzato come ultimo blocco.

Per comprendere al meglio l'algoritmo di consenso proof of stake utilizzato da Tezos, lo suddivideremo in tre sezioni:

- *Creazione di blocchi (cottura),* vengono creati dei blocchi per facilitare il progredirsi della catena; in Tezos, i partecipanti che creano questi blocchi sono chiamati *panettieri*, i quali contribuiscono con la loro potenza di calcolo alla rete per convalidare le transazioni. Per farlo, vengono *ricompensati* dal protocollo sotto forma di XTZ (16 XTZ per blocco). Ma prima di tutto, per essere considerato un panettiere, un partecipante deve possedere almeno 10.000 XTZ (1 rotolo). Più *rotoli* si hanno,

maggiori saranno le probabilità di avere il diritto di cuocere il blocco successivo. Se ci sono 10 rulli attivati ad un certo punto nel tempo, e un panettiere ne possiede 2/10, ha il 20% di possibilità di ricevere i diritti per creare il blocco successivo. Questo significa che se un panettiere ha 10.000 XTZ o 19.999 XTZ, ha gli stessi diritti di cottura nel sistema. I diritti di cottura solo fissati in termini di *priorità* che viene decisa in modo random. Il controllo viene tenuto da colui che ha la priorità 1 ma se non crea e non trasmette un blocco entro un minuto si passa oltre ed il controllo passa a colui che aveva la priorità 2 perché adesso prende il posto della priorità 1 e così via. Più rotoli si possiedono, maggiori sono le probabilità di avere la massima priorità. Per poter cuocere, il panettiere deve versare un *deposito cauzionale (la proof of stake)* di 512 XTZ per blocco creato. Questo deposito è bloccato per cinque cicli (circa 14 giorni) ma se il panettiere cuoce due volte il deposito può essere tagliato. Inoltre, c'è un altro modo per cuocere i blocchi senza allestire un'infrastruttura informatica, cioè delegando le proprie monete ad un *fornaio*. La *delega* permette ai possessori di monete di "prestare" le loro monete ad un fornaio. In questo caso, il fornaio ha una maggiore probabilità di essere selezionato e a sua volta condivide le entrate aggiuntive con il portamonete. Durante questo processo è importante sapere che non vengono trasferite delle monete, infatti i fornai non possono spendere le XTZ che gli

sono state delegate e inoltre non possono scappare con i soldi di altre persone.

- *Regola di scelta a forcella,* l'ultima cosa chiave da capire per comprendere l'algoritmo del consenso di Tezos è come il protocollo decide quale *forchetta a catena* è quella "corretta". La regola di scelta della forchetta di Bitcoin è semplice, la catena più lunga è quella canonica. In Tezos, invece, si sceglie la catena canonica in base al numero di panettieri che hanno approvato il blocco. È stato detto sopra che i panettieri hanno il diritto di creare blocchi, ma oltre a questo hanno anche il diritto di avallare i blocchi. Ad ogni altezza del blocco, 32 rotoli casuali sono selezionati per approvare un blocco, e il blocco con maggior numero di approvazioni è tratto come quello canonico e successivamente il panettiere che ha validato il blocco ottiene la sua ricompensa in XTZ. Quindi i panettieri sono incentivati a sostenere i blocchi che ritengono validi e che altri panettieri potrebbero sostenere.

Riassumendo, il protocollo di Tezos utilizza un algoritmo Proof of stake a catena, in base al quale le approvazioni vengono utilizzate per classificare le catene e decidere qual è quella canonica. I panettieri (persone che possiedono 10.000 XTZ) hanno la responsabilità di creare e avallare i blocchi e inoltre sono tenuti a investire parte del proprio capitale al fine di incentivare un comportamento onesto. Per quanto concerne il problema del *Nothing-at-Stake*, il protocollo Tezos include alcune condizioni di taglio. I panettieri che cuociono o sostengono blocchi multipli della stessa

altezza, perdono il loro deposito di sicurezza. Se qualcuno osserva un altro panettiere che *"cuoce due volte",* può includere un'accusa in un futuro blocco contenente le prove. Tutto questo farà sì che il *"doppio panettiere"* perda il deposito cauzionale e le ricompense future fino a quel punto del ciclo. La metà di questo viene bruciata, mentre l'altra metà va all'accusatore sotto forma di *premio di blocco*. Questo porta i panettieri a tenere sotto controllo gli altri panettieri e ad accusarli quando osservano una doppia cottura, in questo modo i panettieri non cuoceranno o appoggeranno i blocchi su forchette multiple proprio perché il rischio di perdere tutto prevale. Nell'attuale protocollo Tezos, bisogna avere 30 conferme (circa 30 minuti) per considerare una transazione definitiva. Poiché Tezos utilizza un algoritmo di consenso, la possibilità di una riorganizzazione della catena rimane anche dopo una transazione. Gli utenti devono attendere una serie di conferme prima di poter essere assolutamente sicure che una transazione non verrà annullata. Un determinato attore è in grado di riorganizzare un dato blocco solo se controlla l'X% dei rulli.

3.2. Cos'è l'auto-emendamento?

Tezos è una rete a catena di blocchi che si *auto-modifica* e che incorpora un meccanismo a catena per proporre, selezionare, testare e attivare gli aggiornamenti del protocollo senza la necessità di un hard fork. Tutto ciò significa che la blockchain di Tezos può migliorarsi nel tempo avendo un processo formalizzato per gli aggiornamenti del protocollo. Tutto ciò è simile alla struttura di una società, dove gli azionisti possono votare sulla direzione dell'azienda. Molte altre blockchain non hanno questo tipo di struttura formale di governance, quindi le decisioni sono prese da un piccolo gruppo di sviluppatori o da una fondazione che può rappresentare tutti gli stakeholder in modo equo. Il processo di *auto-emendamento* è suddiviso in quattro periodi:

1. *Periodo di proposta*: il processo di modifica di Tezos parte con il periodo di presentazione delle proposte, durante il quale i panettieri possono presentare proposte in successione, quindi il panettiere presenta l'hash del codice sorgente. In questo periodo si possono presentare fino a 20 proposte e la presentazione di una proposta conta anche come voto, il quale nel suo bilancio di puntata all'inizio del periodo, equivale al numero di rulli. Al termine di questo periodo, la rete conta i voti delle proposte e la più votata passa al periodo di votazione per l'esplorazione. Se non ci sono state proposte, oppure se c'è un nesso tra le varie proposte, ne inizia uno nuovo;
2. *Periodo di voto di esplorazione*: in questo periodo i panettieri possono votare sulla proposta più votata del precedente periodo di

proposta. I panettieri possono *votare* "*sì*", "*no*" oppure si possono "*astenere"* dal voto, ma questo significherebbe solo "non votare" su una proposta. Inoltre, come nel periodo di proposta, il voto di un panettiere si basa sul numero di rulli nel suo bilancio di puntata. Al termine della votazione la rete conta i voti, e se il totale di "si", "no" e "astenuti" raggiunge l'obiettivo, e una maggioranza dell'80% dei panettieri non astenuti approva, la proposta procede al periodo di prova. L'obiettivo di partecipazione al voto cerca di eguagliare la media mobile esponenziale del tasso di partecipazione passato. Se la partecipazione al voto non riesce a raggiungere l'obiettivo o la maggioranza dell'80%, il processo di modifica riparte all'inizio del periodo di proposta;

3. *Periodo di prova*: se la proposta viene accettata durante il periodo di votazione di esplorazione, il periodo di prova inizia con una forchetta di prova che funziona in parallelo alla rete principale per 48 ore. Queste forchette hanno accesso alla libreria standard, ma sono fornite anche di sandbox. Questo periodo di test viene usato per determinare se una proposta è un emendamento degno del protocollo. Per non corrompere la rete a catena di blocchi viene utilizzato il *testnet fork,* ma se l'aggiornamento fosse adottato la rete continuerebbe a rendere valide le transazioni di stato;
4. *Periodo di voto*: concluso il periodo di prova, inizia il periodo di votazione promozionale, dove la rete decide se adottare l'emendamento sulla base di discussioni fuori catena e del suo comportamento durante il periodo di test. Come nel periodo di voto

di esplorazione, i panettieri inviano i loro voti utilizzando l'operazione di voto, con i loro voti ponderati secondo il numero di rulli nel loro bilancio di puntata. Terminato quest'ultimo periodo di votazione per la promozione, la rete conta il numero di voti, e se il tasso di partecipazione raggiunge il numero minimo e una maggioranza dell'80% di panettieri non astenuti vota "si", allora viene attivata la proposta come nuova rete principale, altrimenti il processo ritorna di nuovo al periodo di proposta. Il tasso minimo di partecipazione al voto è fissato in base ai tassi di partecipazione passati.

Ognuno di questi quattro periodi ha una durata di otto cicli di cottura (vale a dire 32.768 blocchi o circa 22 giorni, 18 ore), che comprendono esattamente tre mesi dalla proposta all'attivazione.

3.3. Gli smart contracts

Le piattaforme di contratti intelligenti complete Turing come Tezos o Ethereum consentono di eseguire il codice arbitrario in modo affidabile e minimizzato. Tuttavia, alcune applicazione possono essere adatte a Tezos sulla base di una governance formale e incentrata sulla sicurezza degli smart contracts; vediamo alcuni esempi:

- *Attività digitali*: per quanto riguarda Tezos, sono particolarmente adatti dei beni come il denaro digitale, immobili, oggetti da collezione digitali, e così via. Evitare le fork controverse può preservare il valore e il coordinamento all'interno di una rete, rendendo Tezos una piattaforma avvincente per l'emissione di risorse digitali. Sebbene nessun sistema possa essere incondizionatamente sicuro, il linguaggio del contratto intelligente Tezos, Michelson, è stato progettato tenendo conto della sicurezza e della verifica formale;
- *Contratti finanziari minimizzati e affidabili*: sono quei contratti finanziari come scambi decentralizzati, swap, prestiti e così via, i quali richiedono un alto livello di correttezza. Le reti a catena di blocchi decentralizzate traggono il loro valore dall'assenza di un terzo di fiducia, il che rende la perdita di fondi dovuta a un bag nel codice particolarmente spietato.

Il linguaggio di Michelson, utilizzato per scrivere gli smart contracts, è specifico del dominio sulla catena di blocchi di Tezos, e se implementata correttamente, dimostra matematicamente la correttezza del codice,

aumentando la sicurezza dei contratti intelligenti più sensibili o ponderati finanziariamente e riducendo la probabilità di bug. Michelson è un linguaggio basato su *stack*, e non ha alcuna variabile. Questa tipologia di linguaggio opera su uno o più stack e ognuno può avere uno scopo diverso. Le tecniche di verifica formale sono utilizzate in infrastrutture software *mission-critical*, come aeromobili, reattori nucleari e veicoli automobilistici. **Liquidity** è il linguaggio di alto livello utilizzato per programmare *"smart contracts per Tezos".* Esso utilizza la sintassi *OCaml,* si tratta di un linguaggio funzionale e completamente digitale, che rispetta le restrizioni di sicurezza di Michelson. Il linguaggio di Liquidità copre già il 100% delle caratteristiche Michelson, e i contratti generati possono essere inviati sulla rete principale e su zeronet[19]. Per riuscire a dimostrare la correttezza di questo nuovo linguaggio usato per contratti intelligenti, gli sviluppatori stanno lavorando ad un *framework* di metodo formale. La differenza tra *Michelson* e *Liquidity* è data dal fatto che: Liquidity è compilata rigorosamente da Michelson ma la Liquidity è un linguaggio molto più facile da affrontare per molti sviluppatori, in quanto non usa una sintassi estremamente complicata, ma tipi di alto livello piuttosto che manipolazioni di stack. **OCaml** è il linguaggio del protocollo Tezos, un linguaggio di programmazione "*general purpose industrial-strength"* con enfasi su espressività e sicurezza. Questa tipologia di linguaggio è la preferita dalle aziende dove la velocità è cruciale e un singolo errore può costare milioni di euro. Ha una libreria

[19] *ZeroNet* è una rete decentralizzata che ha sede a Budapest (Ungheria), è completamente open source ed è programmata in Python. Inoltre, non è completamente anonima, la rete utilizza una "crittografia a Bitcoin" per tenere al sicuro i dati degli utenti.

standard di alto livello, il che lo rende utile per molte applicazioni di Python o Perl, inoltre ha costrutti di programmazione modulare e orientata agli oggetti che lo rendono applicabile per l'ingegneria del software su larga scala. *OCaml* è anche utilizzato da molte aziende conosciute, tra cui Facebook, Bloomberg, Jane Street e Docker. I punti di forza di questa lingua OCaml sono:

- *Un sistema di tipo potente,* dotato di polimorfismo parametrico e inferenza di tipo, ad esempio: il tipo di una collezione può essere parametrizzato dal tipo di elementi che la compongono. Questo permette di definire alcune operazioni su una collezione indipendentemente dal tipo dei suoi elementi: l'ordinamento di un array ne è un esempio;
- *Tipi di dati algebrici e pattern-matching,* in cui nuovi dati possono essere definiti come combinazioni di record e somme. Le funzioni che operano su tali strutture di dati possono quindi essere definite dal pattern matching, una forma generalizzata della nota dichiarazione di commutazione, che offre un modo pulito ed elegante di esaminare e denominare i dati in modo simultaneo;
- *Gestione automatica della memoria,* grazie ad un discreto, incrementale e veloce sistema di raccolta rifiuti;
- *Compilazione separata di applicazioni autonome,* in cui i compilatori bytecode portatili permettono di creare applicazioni autonome dai programmi *Caml Light* o *OCaml*. Un'interfaccia di funzione esterna permette al codice OCaml di interagire con il

codice C quando necessario. L'utilizzo interattivo di OCaml è anche supportato da un ciclo di *"lettura-valutazione-stampa".*

Inoltre, questo linguaggio è dotato di alcune funzionalità:

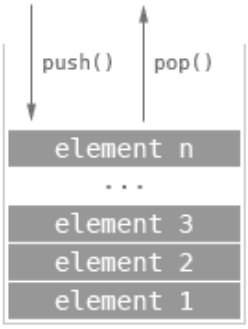
- *Un sofisticato sistema di moduli,* che permette di organizzarli in modo gerarchico e di parametrizzarli rispetto ad altri moduli;
- *Un livello espressivo orientato agli oggetti,* con classi multiple di ereditarietà, parametriche e virtuali;
- *Efficienti compilatori di codice nativo,* oltre al compilatore di bytecode, OCaml offre un compilatore che produce codice macchina efficiente per molte architetture.

Inizialmente il linguaggio OCaml è stato utilizzato per sviluppare applicazioni che implicano il calcolo simbolico, ora invece è utilizzato per sviluppare software in molte altre aree applicative.

3.3.1. Il linguaggio di Michelson

Come è stato detto precedentemente, per scrivere gli smart contracts, Tezos utilizza il linguaggio di programmazione a basso livello chiamato Michelson e basato su *stack*. Progettato principalmente per facilitare la verifica formale e per consentire agli utenti di dimostrare le proprietà dei loro contratti, inoltre, utilizza un paradigma di ristrutturazione dello stack, per cui ogni funzione riscrive uno stack di input in uno stack di output. Esso è un tipo di dato astratto che serve come una raccolta di elementi, con due operazioni principali: **push** (aggiunge un elemento alla collezione) e **pop** (rimuove l'ultimo elemento aggiunto alla collezione e non ancora rimosso), **LIFO** (last in, first out) è il nome che identifica l'ordine in cui escono da una pila gli elementi.

Per riscrivere gli stacks effettuando una transazione in Michelson, prima di



tutto lo stato della catena di blocco ad un certo hash viene decentralizzato e messo in pila come *storage* (memoria) variabile, quindi abbiamo una *from* funzione che riceve la transazione *amount* (quantità di dati), la quantità di dati allegati ꜩ (tez), e il *parameter* (parametri) della funzione. Dopo l'esecuzione della funzione, senza alcun aggiornamento dello stack, il programma chiamerà una to funzione che ha il *result* (risultato) dei parametri, che è il risultato della funzione, e l'output *storage* che è

serializzata e memorizzata sulla blockchain. In questo esempio, Michelson manipola lo stack solo funzionalmente e un nuovo stack viene passato da funzione a funzione. Secondo *Milo Davis* l'utilizzo di Michelson per gli smart contracts in Tezos spesso viene visto come una "lingua strana", infatti non include il polimorfismo, le chiusure o le funzioni nominate, quindi rispetto ad un linguaggio come OCaml, sembra poco potente. Inoltre, il suo stack non è sempre facile da gestire, e non c'è una libreria standard; tuttavia, queste restrizioni sono in gran parte motivate dagli obiettivi di progettazione della lingua. Abbiamo due motivazioni per cui il linguaggio di Michelson viene mantenuto:

- *Prima di tutto per fornire un bytecode leggibile;*
- *Per essere introspettabile.*

Inoltre, gli smart contracts li ritroviamo anche in Ethereum; Tezos, però, ne ha una visione leggermente diversa. Per esempio, pensiamo alla piattaforma più come un modo per implementare alcuni pezzi di logica di business che come un generico "world computer": in Ethereum, la maggior parte degli smart contracts implementa cose come portafogli multisig, regole di vesting e distribuzione, e **Michelson** si rivolge a questo tipo di applicazioni. Michelson è un linguaggio che può essere scritto a mano, nonostante fosse stato progettato come *target di compilazione leggibile.* L'obiettivo principale è che anche l'output di un compilatore possa essere compreso. Intendiamo che il linguaggio sia abbastanza semplice da permettere agli sviluppatori di costruire i propri strumenti di analisi. Tutto questo, però, è un allontanamento dal bytecode **dell'EVM** *(Ethereum Virtual Machine)*, infatti con un bytecode di livello inferiore

spesso non si riesce a risalire alle proprietà del programma effettivamente eseguito, mentre con Michelson tutto questo è molto più facile. L'utilizzo di un bytecode di livello superiore semplifica anche il processo di verifica della proprietà dell'output compilato. I programmi scritti in Michelson possono essere analizzati senza la necessità di tecniche più complicate come la logica di separazione. L'attuale implementazione di Michelson si basa su un OCaml GADT[20], che abbiamo usato per verificare la sonorità del linguaggio. Inoltre, l'implementazione di un linguaggio basato su stack mappa direttamente alla semantica. Infine, uno dei principali vantaggi di Tezos è che il sistema è modificabile, infatti possiamo iniziare con un piccolo linguaggio di base in cui siamo fiduciosi e aggiungiamo funzionalità man mano che vengono creati casi di buon uso per loro. In riferimento a tutto questo *Olin Shivers* dice: "si dovrebbe sempre usare uno *strumento abbastanza piccolo per il lavoro*", a tal proposito Michelson è stato progettato proprio per risolvere questa richiesta.

[20] GADT sta per Tipi di dati algebrici generalizzati i quali consentono agli sviluppatori di OCaml di descrivere ricche relazioni tra i costruttori di dati e i tipi in cui vivono. Attualmente il linguaggio Michelson utilizza GADT per la verifica formale dei tipi.

3.3.2. Il linguaggio Liquidity

Per programmare smart contracts per Tezos viene utilizzato il *linguaggio di programmazione Liquidity,* un linguaggio funzionale completamente digitato. *Liquidity* usa la sintassi di OCaml e rispetta le restrizioni di sicurezza che impone Michelson. Attualmente gli sviluppatori stanno lavorando ad un framework di metodo formale che verrà utilizzato per dimostrare la correttezza degli smart contracts scritti in Liquidity. Le caratteristiche di questo linguaggio sono le seguenti:

- *Copertura completa del linguaggio Michelson*: tutto ciò che può essere scritto in Michelson è scritto anche in Liquidity;
- *Variabili locali invece di manipolazioni di stack:* i valori possono essere memorizzati in variabili locali;
- *Tipi di alto livello*: i tipi come *tipi di somme* e *tipi di record* possono essere definiti e utilizzati nei programmi di Liquidity;
- *Un modulo e un sistema di contratto per scrivere codice e librerie riutilizzabili;*
- *Un potente meccanismo di inferenza di tipo con polimorfismo;*
- *Una sintassi ReasonML*[21] alternativa per scrivere contratti in un linguaggio simile a Javascript;
- *Un compilatore efficiente e ottimizzante;*

[21] *ReasonML* non è una nuova lingua ma una nuova sintassi di concatenamento alimentata dal linguaggio testato in battaglia, *OCaml*. Reason fornisce a OCaml una sintassi familiare, orientata verso i programmatori JavaScript, infatti può essere considerato come un cugino, molto più veloce e semplice.

- *Un decompilatore del programma Michelson per quelli di Liquidity:* tutti i programmi Michelson possono essere decompilati in una forma leggibile.

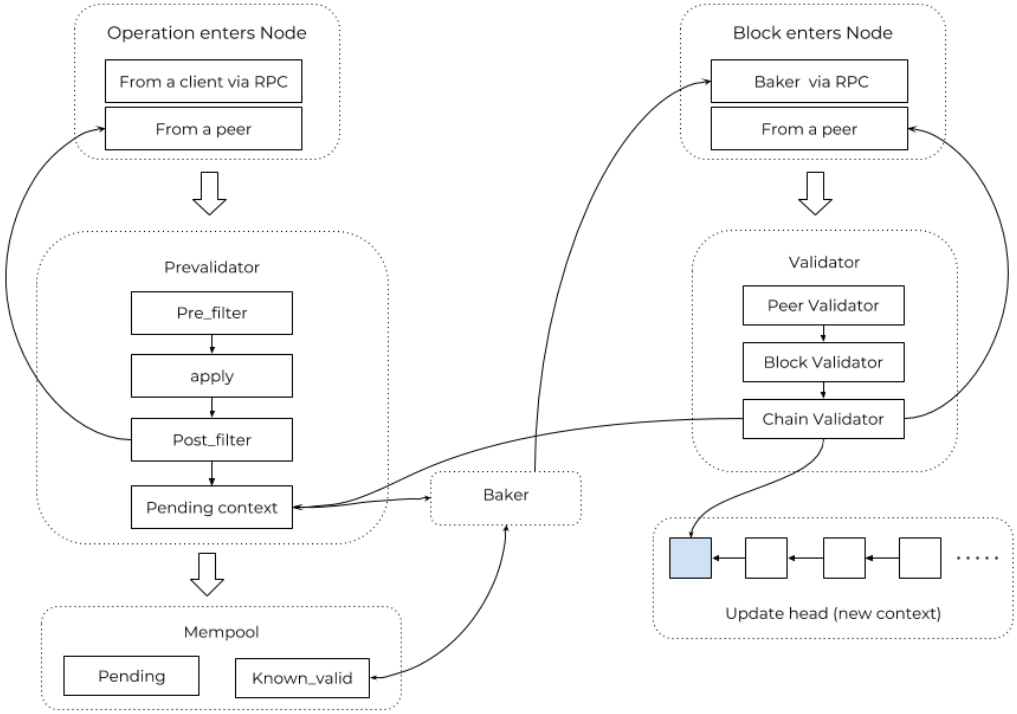
L'idea di creare un linguaggio intuitivo per gli smart contracts è nata nell'estate del 2017 e una prima versione è stata rilasciata proprio in quello stesso anno. Il team ha fatto molti sforzi per migliorare l'intero framework, che purtroppo la **Foundation di Tezos** ha deciso di non supportare, infatti lo sviluppo di Liquidity è ora supportato da Dune Foundation[22] e Origin Labs[23].

[22] Dune Foundation è una piattaforma Web 3.0 protetta e progettata per applicazioni decentralizzate di livello aziendale. Dune sta rendendo la sicurezza accessibile fornendo una gamma di lingue native per contratti intelligenti senza compromettere le prestazioni, tratto da: "https://dune.network/".

[23] Origin Labs è una piattaforma decentralizzata che ha una vasta esperienza e competenza nella costruzione di blockchain e applicazioni, tratto da: "https://origin-labs.com/".

3.4. Il ciclo di vita di un'operazione

Il ciclo di vita che compie un'operazione dal momento in cui entra in un blocco fino all'inclusione nella blockchain di tezos viene rappresentato da questo schema:



Inoltre, le operazioni vengono ricevute dai nodi in due modi diversi:

- *Da un client tramite RPC;*
- *Da un peer tramite pubblicità.*

Tutte le operazioni locali sono ricevute dai *servici RPC* esposti dalla shell, quindi queste operazioni passano attraverso il sottoinsieme di convalida. Il nodo mantiene un pool di memoria ("*mempool*") per tenere traccia delle operazioni non valide, esse sono due:

- *Conosciuto_valido (Known_valid),* è una lista di operazioni valide da applicare al contesto corrente (stato) e che possono essere incluse in un blocco se richiesto da un fornaio. Il contesto risultante dalle operazioni è chiamato *"contesto in sospeso";*
- *In attesa (pending),* un insieme limitato di operazioni che, come è noto, non sono sicuramente non valide e possono essere trasmesse ai *peer*. Questo insieme contiene due tipi di operazioni:
    - *Ramo_rifiutato (Branch_refused):* operazione che potrebbe essere valida in un ramo diverso;
    - *Ramo_ritardato (branch_delayed):* operazione arrivata troppo presto.

Ogni volta che il *mempool* viene aggiornato, le operazioni vengono trasmesse ai coetanei, inoltre il nodo peer recupera l'operazione dal peer remoto, utilizzando l'hash dell'operazione. In questo modo un'operazione entra in un nodo e tutte passano attraverso il sottosistema di validazione. Quando un'operazione viene ricevuta da un peer, viene notificato al nodo peer, il risultato del passaggio di validazione. Un'operazione viene rimossa da mempool solo se nel corso del suo tempo di vita non viene aggiunta a un blocco valido della catena che il nodo considera canonica. Il responsabile che decide quale operazione deve essere aggiunta al mempool è il *prevalidator nel sottosistema di validazione.* Grazie allo schema visto sopra, possiamo comprendere che un'operazione attraversa i seguenti passi: *pre_filtro*, *applica* e *post_filtro* all'interno del prevalidator; il fallimento in qualsiasi passo comporta il rifiuto della transazione:

- La funzione del *pre_filtro* esegue alcuni controlli di base per le operazioni di gestione, tra cui il gas, assicurandosi che il mittente abbia un saldo sufficiente (in termini XTZ) per pagare il gas;
- Nella seconda fase, *l'applicazione*, tutte le operazioni che superano il pre_filtro vengono applicate al contesto in sospeso. In caso di esito positivo, la ricevuta dell'operazione e il nuovo contesto in sospeso verranno ritrasmessi, mentre in caso di guasto viene trasmessa una traccia di errore. Le operazioni che causano errori permanenti vengono rifiutate, le altre vengono inserite nell'elenco delle operazioni in sospeso del mempool. Ad esempio, un *errore permanente* potrebbe essere: l'esaurimento del gas, oppure l'attivazione non valida o prove incoerenti in caso di denuncia come prove di doppia cottura o prove di doppia girata; per quanto riguarda *errori temporanei* potremmo avere: il periodo di voto in una scheda elettorale non è il periodo di voto corrente, o la prova della doppia cottura è ricevuta troppo presto; infine, un esempio di un errore di ramo è una prova di doppia cottura non richiesta quando il delegato è già denunciato nel ciclo corrente;
- Nella terza ed ultima fase, il *post_filtro*, viene presa la decisione di aggiungere un'operazione al mempool in base al risultato. Questo viene fatto tramite la funzione post_filter. Tutte le operazioni aggiunte vengo propagate tramite la pubblicità e questa propaganda pubblicizza il tutto utilizzando *current_head* di *distributed_db* nel modulo di pubblicità.

Quando un fornaio deve cuocere (produrre) un blocco, interroga il mempool per ottenere tutte le operazioni valide. Terminato il tempo e raggiunto il limite, un fornaio fa un blocco con le conferme che ha ricevuto e successivamente i blocchi vengono iniettati nei nodi tramite una chiamata *RPC* utilizzando il modulo di iniezione in *lib_shell_services.* Se un blocco viene iniettato da un fornaio, viene chiamato direttamente poiché i dati richiesti per l'avvio di una validazione di un blocco sono già presenti localmente, altrimenti, se viene recuperato da un peer, tutti i dati necessari per convalidare il blocco vengono recuperati dal *validatore peer* usando il db distribuito. Il *validatore di blocchi* convalida un blocco e chiama il *validatore di catene* se l'attuale capo della catena può essere aggiornato; questa funzione convalida il blocco utilizzando *l'apply_block.* La *convalida di intestazione* include controlli per errori come:

- *livello non valido*;
- *non crescente timestamp;*
- *non aumentando il fitness*;
- *inaspettato numero di passaggi di validazione*.

La *convalida del blocco* include controlli per errori come troppe operazioni, operazione sovradimensionata, versione del protocollo errata, passaggio di convalida non consentito, idoneità non valida, protocollo non disponibile ed errori durante l'applicazione dell'operazione. Convalidato il blocco e candidato per il nuovo capo della catena, viene passato al *validatore della catena,* il quale verifica che il punteggio di fitness della nuova testa sia superiore al punteggio di fitness della testa corrente. In caso contrario, il blocco viene ignorato. Infine, la

nuova testa del blocco viene pubblicizzata sul peer usando il modulo *pubblicizza di distribut_db* e questo blocco diventa parte della catena canonica solo se i futuri fornai ci cuociono sopra, quindi tutte le operazioni all'interno di questo blocco fanno parte della blockchain di Tezos.

3.5. Sicurezza degli smart contracts

Dal punto di vista giuridico, dice *"Kate Sills"*, gli smart contracts non sono mai stati visti come contratti legali. Nick Szabo voleva creare nuove istituzioni digitali: accordi applicati in codice cartaceo piuttosto che tribunali, e ovviamente i tribunali fisici non potevano tenere il passo con Internet. In questo contesto anche Satoshi ha espresso interesse nel sostenere un'ampia gamma di impegni per quanto concerne Bitcoin, ma ancora oggi le piattaforme di contratti intelligenti come Tezos seguono il meccanismo di lavoro proposto da Szabo e tutto questo ci consente di assumere impegni con estranei su Internet in codice. In generale, riuscire ad utilizzare gli smart contracts è un tassello senza il quale non si potrebbe vivere in quanto saremmo limitati al baratto simultaneo, ad esempio: un utente ha un oggetto A, un altro ha un oggetto B e commerciano sul posto. Tuttavia, più economisti chiamano la *"doppia coincidenza dei desideri"*: per poter commerciare, un utente deve volere l'oggetto A nello stesso momento in cui l'altro utente vuole l'oggetto B, il che non è molto probabile. Per risolvere questo problema si utilizza il

denaro, cioè si può vendere l'oggetto B e in seguito con i soldi ricavati acquistare l'oggetto A o un altro oggetto. In questo modo, grazie ai soldi si possono ridurre al minimo i "desideri" da due a uno. Questo era solo un piccolo esempio per spiegare in modo semplice i contratti, i quali funzionano in modo simile, ma facilitano ancora di più potenziali transazioni. Qui non è richiesto uno scambio simultaneo di valore, nemmeno di denaro, infatti è possibile vendere l'oggetto B e un altro utente può promettere il rimborso il mese successivo. Tale capacità espande radicalmente i tipi di transazioni che è possibile effettuare. L'unico problema che sorge spontaneamente è: *l'utente manterrà la promessa di rimborso?* Ecco perché nascono i contratti intelligenti per riuscire a mantenere una promessa fatta, quindi nel 2013 il mondo delle criptovalute ha ampliato il concetto di contratti intelligenti. Attualmente sono in corso molti sforzi per migliorare Tezos in termini di privacy, consenso, scalabilità, contratti intelligenti e governance. Inoltre, alcuni sviluppatori stanno esplorando nuovi algoritmi di consenso e questi a loro volta vengono sviluppati da altri team in modo che possano essere inclusi nel protocollo Tezos.

Conclusione

Dal seguente elaborato emerge che la tecnologia Blockchain è una novità rivoluzionaria e presto cambierà molti aspetti del futuro. La sua capacità di trasformare sistemi tradizionali in maniera sicura, trasparente, distribuita e collaborativa ha trasmesso più sicurezza agli utenti facendoli diventare più responsabili nell'utilizzo di questa tecnologia. Il focus è stato incentrato sulle criptovalute, in particolar modo sulla moneta di Tezos che a differenza di Bitcoin ed Ethereum ha qualche particolarità in più, sia dal punto di vista tecnico che dal punto di vista giuridico, infatti il fondatore di Tezos, *Arthur Breitman,* fin dall'inizio aveva notato "l'**incapacità di evolversi***"* di Bitcoin e per questo ha pensato di inventare una nuova moneta che potesse superare quest'ostacolo, ecco come si arriva alla nascita di Tezos. Ethereum, invece, è considerata la criptovaluta che più si avvicina a Tezos, infatti risulta che siano molto simili tra di loro e si differenziano solo per alcuni dettagli. Altro punto fondamentale su cui si è discusso sono gli **smart contracts**, sia dal punto di vista della moneta di Ethereum che di Tezos, andando a vedere nei dettagli le varie tecnologie per il linguaggio di programmazione utilizzato; i meccanismi di svolgimento per formare i blocchi della blockchain; il punto di vista della sicurezza e della privacy. Tutto questo ci ha portato a capire che molto probabilmente Tezos potrebbe essere considerata migliore di altre criptovalute soprattutto per il fatto che sia auto-modificante, privilegio che le altre criptovalute non hanno.

FONTI BIBLIOGRAFICHE E SITOGRAFIA

HashCash e Bitgold:

https://cryptonomist.ch/2019/01/12/moneta-digitale-prima-di-bitcoin-hashcash-e-bitgold/

La blockchain:

https://arxiv.org/pdf/1904.00315.pdf

https://allquantor.at/blockchainbib/pdf/abeyratne2016blockchain.pdf

La crittografia:

https://www.criptoinvestire.com/come-funziona-la-crittografia-nelle-blockchain.html

La criptovaluta di Bitcoin:

https://nexa.polito.it/nexacenterfiles/lunch-20-giobergia.pdf

https://www.binance.vision/it/blockchain/what-is-bitcoin

La criptovaluta di Ethereum:

https://www.binance.vision/it/blockchain/what-is-ethereum

https://arxiv.org/pdf/1901.01376.pdf

https://arxiv.org/pdf/1908.11808.pdf

https://github.com/ethereum/wiki/wiki/White-Paper#ethereum

La moneta di Tezos:

https://tezos.com/static/white_paper-2dc8c02267a8fb86bd67a108199441bf.pdf;

https://tezos.com/get-started

https://medium.com/tqtezos/lifecycle-of-an-operation-in-tezos-248c51038ec2

Le criptovalute:

https://www.binance.vision/it/blockchain/what-is-cryptocurrency

Storia della Blockchain:

https://legacyshop.wki.it/documenti/00240155_est.pdf?download=true

Vantaggi e svantaggi della blockchain:

https://www.binance.vision/it/blockchain/positives-and-negatives-of-blockchain

RINGRAZIAMENTI

Con questa tesi giunge ufficialmente al termine il mio percorso universitario: non è stato facile arrivare fin qui, tuttavia eccomi.

Colgo l'occasione, prima di tutto per ringraziare il prof. Sangiorgi Davide per la fiducia accordatami accettando il ruolo di Relatore, per la sua disponibilità e per avermi aiutata a capire come sviluppare il mio elaborato nel migliore dei modi.

Il ringraziamento più importante va ai miei genitori, mamma Antonella e papà Gennaro che, con il loro sostegno, sia morale che economico, mi hanno permesso di arrivare a Bologna per intraprendere questo percorso formativo, a Marco e Francesca, mio fratello e mia sorella, per le "tante chiamate" che non mi hanno fatto in questi anni, nonostante tutto vi voglio bene.

Un ringraziamento speciale va anche a tutta la mia famiglia che si è sempre interessati al mio percorso universitario, i miei zii e tutti i miei cugini, in particolar modo volevo ringraziare mio cugino Alessandro ed Emanuela che oltre ad avermi ospitata il primissimo anno in cui sono arrivata a Bologna, mi hanno sostenuta e incoraggiata fin dall'inizio; mia cugina Alessandra per le tante ore passate al telefono a farci compagnia a vicenda, le mie due nonne: nonna Teresa che, nonostante tutto, riesce sempre a chiamarmi, anche solo per chiedermi come va; e nonna Angela, la quale nonostante sia venuta a mancare a metà del mio percorso formativo, ricordo ancora le sue chiamate ogni sabato pomeriggio.

Il ringraziamento più caloroso va a Carmine, il mio fidanzato, che è riuscito a supportarmi e sopportarmi senza mai farmi pesare nulla, standomi vicino soprattutto nei momenti peggiori quando ero veramente insopportabile,

ringrazio anche i suoi genitori per le tante cose buone da mangiare portatemi dalla Puglia.

Volevo ringraziare anche tutti i miei amici di giù, in particolare Anna per il suo supporto morale anche se a distanza; Francesco (Ciccio) che nonostante i suoi ripetuti messaggi chilometrici con scritto: "Giadaaaaaa studiaaaaa", è riuscito sempre a strapparmi un sorriso; e poi i miei amici di Bologna con i quali ho condiviso quest'esperienza facendomi sentire sempre a casa, principalmente Chiara la mia primissima coinquilina sempre pronta a sostenermi e aiutarmi in tutto; Stefania con la quale abbiamo passato momenti memorabili, ridendo, bevendo e immaginando di essere al mare quando in realtà eravamo in studentato a studiare; Sabrina sempre pronta a preparare da mangiare per assicurarsi che nessuno in casa fosse rimasto a digiuno; Alessandra, collega di università, con la quale ci siamo fatte tante risate a lezione, nonostante i suoi ripetuti ritardi lei c'era sempre, e infine tutti gli altri amici che hanno reso questi 4 anni fantastici e indimenticabili, grazie per aver condiviso con me quest'esperienza così importante.