DAVIDE CARBONI

DAGLI SMART CONTRACT ALLE ICO

LA BLOCKCHAIN NON DORME MAI

Contents

DAGLI SMART CONTRACT ALLE ICO

Davide Carboni

Pubblicato da immutable.today

Presentiamoci

A chi è rivolto

Questo libro è rivolto a te prima che il gruppo mamme della scuola ti chieda di partecipare ad una ICO per la fiera di Natale quest'anno.

E' rivolto a chi ha solo sentito parlare di Bitcoin ma ha scoperto che esistono anche Ripple, Ethereum, Dash e

Dogecoin, una buffa moneta elettronica con un cane meme come mascotte.

E’ rivolto al papà che lo vuole spiegare al figlio adolescente, oppure al figlio adolescente che lo vuole spiegare al papà. Insomma fate voi.

E’ rivolto a chi non ne ha mai sentito parlare e gli capita questo libro fra le mani per sbaglio o perché gli piace la copertina. Beh, non dovrebbe essere così male come primo approccio all’argomento.

E’ rivolto a chi sa già tutto e vuole solo mettere una stella su Amazon. Grazie lo stesso. Parliamone.

Viviamo ormai in un'epoca in cui le innovazioni tecnologiche sono così tante che quasi non ci sorprendono più. E' giusto trascurare ciò che non ci interessa da vicino. Il mondo offre perfino troppe occasioni di distrazione mentre il nostro tempo è così limitato e così prezioso. Quindi filtrare, selezionare, andare al succo diventa indispensabile. Questo è anche il mio mestiere, informarmi sulle novità tecnologiche ma separare il valore dalla "fuffa".

Ricordiamoci di non sopravvalutare una nuova tecnologia nel breve periodo e di non sottovalutarla nel

lungo periodo.

Chi sono io?

Se hai già letto il mio “I bitcoin sotto il materasso” allora mi conosci già e puoi saltare questa sezione. Sei benvenuto comunque ad una seconda lettura se proprio insisti. Proviamo a descrivere l’autore, cioè il sottoscritto, attraverso una lista di dieci “fatti”.

Davide si interessa di tecnologia da sempre. Ispirato dai robottoni giapponesi è affascinato dai severi e disciplinati scienziati che li costruiscono più che dai piloti belli e ribelli che li guidano in battaglia. Trova tuttavia in Tony Stark una sintesi perfetta capace di ottenere accettazione sociale e successo sentimentale.

In virtù di questa ispirazione, mentre i suoi compagni delle elementari vogliono tutti diventare degli astronauti lui vorrà diventare “uno di quelli che inventano i robot”. Sfortunatamente non costruirà mai un robot ma studierà fisica, matematica, statistica, elettronica, informatica per tutta la vita.

Vive in pieno la rivoluzione Internet, ma forse si trova un po' troppo giovane quando è il momento di cavalcarla e troppo anziano quand'è il momento di inseguirla. O almeno questa è la scusa che racconta a sé stesso.

Nel 2010 ospite di un convegno sul nascente Internet delle Cose, scopre che senza saperlo è un esperto del tema almeno dal 2006. A seguito di questa presa di coscienza inizia un percorso che lo porterà a fondare qualche anno dopo e insieme a tre colleghi la start up paraimpu.com

Nel 2012 si imbatte nel Bitcoin e lo

declassa a “mica posso seguire ogni diavoleria dei nerd”.

Nel 2013 ci ripensa ma è tardi, un Bitcoin vale già 30$ - chi può essere così pazzo da comprare dei fagioli magici a 30$ l’uno? - quindi non diventerà ricco grazie ai Bitcoin. Non rassegnandosi all’idea cerca di capirne il più possibile e studia tutto quello che gli capita sottomano sull’argomento e sì … non c’è niente da fare, troppo tardi per diventare ricco grazie ai Bitcoin.

Nel 2016 scrive “I bitcoin sotto il materasso”. Se fossi bravo a vendere me stesso e quello che faccio lo definirei “un best seller”. Mi accontento di dire

che il libro mi ha dato qualche soddisfazione, prima fra tutte le reazioni dei lettori che mi hanno premiato con i loro commenti e i loro apprezzamenti.

Sono solo otto cose fin qui. Davide ha imparato che a volte bisogna aggiustare gli obiettivi perché non si possiede mai tutta la conoscenza necessaria prima di iniziare un'impresa piccola o grande che sia.

Ma smettiamola di parlare in terza persona e raccontiamo qualche fatto sostanziale.

Sono nato a Cagliari e qui ho studiato e lavorato per gran parte della mia vita. Ho una laurea in Ingegneria

Elettronica guadagnata all'Università di Cagliari ed un dottorato in Informatica, questo però ottenuto all'Università di Sherbrooke in Quebec. Ho iniziato a interessarmi di informatica all'età di 13 anni con l'avvento dei computer domestici. Prima Commodore 64, ZX Spectrum e qualche anno dopo Amiga e finalmente PC (compatibile) IBM.

Ho lavorato in progetti legati alla tecnologia per tutta la mia carriera, in collaborazione o alle dipendenze di aziende e centri di ricerca come il CRS4 di Cagliari, il laboratorio IMS di Bordeaux, l'Intel Collaborative Research Institute a Londra. Come già accennato, ho co-fondato la startup

paraimpu.com nel 2013 e ho anche insegnato per tre anni come professore a contratto per il corso di Ingegneria delle Telecomunicazioni dell'Università di Cagliari.

Tutto questo fa di me un esperto di Bitcoin e blockchain al punto tale da giustificare l'arroganza di scrivere un libro niente meno che sulla nascente cripto economia? Non questo forse, la mia formazione sicuramente mi ha portato ad essere più di altri affascinato da questo tema, ma in realtà sono sicuro che là fuori ci sono commercialisti, avvocati, e laureati in teologia che conoscono questi temi meglio di me. Tuttavia ho sperimentato personalmente

tutte le cose che racconto, non mi invento niente e non consiglio mai cose che non ho provato sulla mia pelle. Ho speso soldi miei per acquistare bitcoin su MtGox. Ho perso i miei bitcoin prestandoli su BTCJam. Ho guadagnato (pochi) bitcoin con il *mining*. Li ho acquistati a 60$ per rivenderli a 500$ pensando che più di così non potessero salire per poi vederli schizzare a 1200$ e riacquistandoli a 500$. Ho partecipato a meetup, conferenze, chiacchiere da bar e brainstorming. Ho coinvolto persone, aziende, centri di ricerca, università per fare squadra, ideare e proporre nuove soluzioni e ancora oggi lo faccio, tutti i giorni.

Aggiungo un piccolo *disclaimer*. Come accennato sopra, tutto quello che troverete in questo libro è frutto di esperienze personali o di documentazione raccolta in rete. Le opinioni qui espresse sono mie e non riconducibili a nessuna delle organizzazioni con le quali ho collaborato nel corso della mia carriera. I marchi citati sono dei rispettivi proprietari e vengono citati per semplice esigenza di cronaca.

Aggiungo anche un grande *disclaimer*. In nessun caso quello che leggerete in questo libro va considerato una guida all'investimento finanziario, io non sono il vostro consulente

finanziario. In nessun caso quello che leggerete in questo libro va considerato un consiglio legale, io non sono il vostro avvocato. Per la precisione non sono né un avvocato né un consulente finanziario. Sono un ingegnere e tutto sommato ne vado fiero.

Non è un libro su Bitcoin

Cosa pensereste se vostro figlio vi chiedesse di inviare la paghetta a 1DchDwds8Tpgxw1paCY4eRSskx8Q53 Beh, che forse vi siete persi inconsapevolmente circa otto anni di una grande rivoluzione tecnologica e finanziaria che inizia come la trama di un film di fantascienza. Nel 2008, nel

bel mezzo della crisi finanziaria che mette in ginocchio l'economia mondiale, pare che uno scienziato anonimo che si fa chiamare Satoshi Nakamoto abbia inventato una nuova forma di moneta digitale. Che lui esista davvero, ad oggi, non è certo. Ma che sia un singolo o un collettivo, quello che crea è un tipo di denaro totalmente gestito da un algoritmo che viene creato dai computer e trasmesso in rete. Un denaro che non può essere censurato, fermato alle frontiere, contraffatto, inflazionato o confiscato. Diciamo che ormai non può più essere trascurato, il Bitcoin.

Questo non è un libro su Bitcoin. E' però impossibile non parlare della

prima - e per alcuni - dell'unica criptovaluta della rete. Nelle prossime sezioni di questo libro cerchiamo di descrivere alcuni concetti fondamentali ma se vuoi saperne di più ti consiglio di partire dalle seguenti risorse:

Nakamoto paper. E' il primo documento che descrive il funzionamento della rete Bitcoin. E' stata pubblicata su The Cryptography Mailing list at metzdowd.com

bitcoin.it è il dominio in cui trovare il wiki che raccoglie molta della documentazione aggiornata sulla rete Bitcoin.

Andreas Antonoupulos. Mastering Bitcoin, Second Edition (Programming the Open Blockchain)

Poi mi faccio un po' di pubblicità: I bitcoin sotto il materasso disponibile su Amazon.it in formato paperback e Kindle.

Un po' però lo è

Capire Bitcoin è difficile. Eppure potremmo restare sorpresi dal fatto che capire il denaro che usiamo tutti i giorni è forse ancora più arduo che capire cos'è e come funziona il Bitcoin. Perché è difficile? Cominciamo dal linguaggio. Una volta nelle banconote italiane, e

ancora oggi nelle banconote inglesi, c'era scritto:

prometto di pagare il Portatore della presente una somma di 5£

E' un linguaggio strano, ma cosa significa? Che se vado dalla regina con la banconota lei mi da 5£? Ma no – direte voi – significa che c'è un sottostante, una riserva aurea e loro ti danno una pagliuzza di oro del valore di 5£. Falso, il denaro non è più garantito dall'oro ormai da tanti anni.

Se andiamo ad esplorare meglio come viene creato il denaro scopriamo che viene generato dal debito.

Praticamente nel momento in cui voi andate in banca a chiedere cento euro, cento euro vengono creati dalla banca. Viene creata una coppia di debito/credito e la ricchezza sottostante a garanzia del denaro è **la volontà del debitore onesto di ripagare il debito** attraverso il lavoro e quindi attraverso la creazione di nuova ricchezza.

Torniamo al Bitcoin e alla sua blockchain. Cos'è la blockchain dunque? Non è solo hype e Bitcoin non è solo la moneta dei criminali, ma è un'invenzione geniale che **ormai c'è e non può essere "de-inventata".**

Quando si parla di sicurezza

informatica bisogna definire il cosiddetto modello di attacco, o modello dell'avversario. Il modello di attacco che ha reso Bitcoin diverso da un generico database è quello di un ambiente ostile in cui **nessuno è autenticato**, nessuno è incaricato di mantenere il server centrale, dove tutti possono accedere con identità multiple e dove **tutti sono potenzialmente intenzionati ad imbrogliare** creando copie non regolari della blockchain per trarne profitto. Questo modello di attacco non può allo stato dell'arte essere sostenuto se non con una blockchain basata sull'invenzione di Satoshi Nakamoto. Se il modello di attacco è più bonario di quello descritto

sopra, allora basta un database transazionale, che in generale è più veloce ed efficiente di una blockchain.

La blockchain oltre il Bitcoin

Siamo all'alba di una nuova economia ed una nuova finanza. Il crowd funding e in generale il concetto stesso di finanziamento delle imprese sta per essere completamente trasformato.

Negli ultimi tre anni dal 2014 con il

lancio di Ethereum (un sistema ispirato a Bitcoin), e fino ai giorni scorsi in cui centinaia di non-company cercano di finanziarsi con una ICO, cioè una "Initial Coin Offering" abbiamo assistito allo svilupparsi impetuoso di questo fenomeno. Gli utenti versano una quota in bitcoin o altre criptovalute e ottengono in cambio altri token digitali che sono registrati dentro una blockchain il cui significato è ancora da chiarire. Per alcuni si tratta di un pericoloso imbroglio, per altri di una poderosa innovazione.

Questi token sono scambiabili e fungibili. In questo modo ogni progetto o ogni azienda può battere la sua moneta

virtuale. Di fatto ormai assistiamo ad una marea di progetti, che invece che andare a cercare finanziamenti su kickstarter o attraverso business angels e venture capital, preferiscono **andare direttamente agli utenti e chiedere un finanziamento in cripto-valute**.

Come detto in precedenza è un territorio completamente nuovo, ricco di opportunità ma anche di truffe perché non regolamentato. Ci sono progetti che hanno mostrato una certa affidabilità, ma ci sono anche tanti altri progetti che si sono dimostrati degli imbrogli, in cui uno investe dei bitcoin e in cambio non ottiene niente.

Va ricordato che nel caso dei bitcoin e delle altre cripto-valute a questo ispirate, le transazioni sono irrevocabili. Se siamo vittime di un raggiro, non c'è autorità di questa terra a cui potersi affidare per ottenere la restituzione di queste somme una volta trasferite. Il motivo è che una volta che una transazione viene sigillata nella blockchain e copiata in migliaia di computer del mondo, nessun giudice potrà più intervenire per modificarla ope legis.

Oltre i semplici servizi di pagamento, servizi più articolati e complessi si possono implementare ed eseguire sulla blockchain, un esempio

sono i cosiddetti smart contract. Se Bitcoin ha introdotto il concetto di criptomoneta, uno smart contract possiamo immaginarlo come **denaro programmabile**. Non immaginate un software che gestisce denaro, ma piuttosto un **denaro che contiene il suo software**.

Gli smart contract sono programmi "unstoppable", cioè nessuno può impedire che il contratto esegua la sua logica perché non esiste un server che può essere chiuso o spento. Ogni smart contract gira in modo identico in migliaia di computer nella rete. Come le teste dell'Idra, chiudendo un server ne rimangono magari altri migliaia in

funzione e quello che è stato chiuso viene rimpiazzato con altri tre.

Lo sviluppo degli smart contract può portare a delle conseguenze interessanti. Per esempio nel campo delle assicurazioni potrebbero nascere dei prodotti completamente nuovi ed automatizzati. Pensiamo ad un assicurato che riceve un risarcimento al verificarsi di un certo evento dove la somma del risarcimento è già depositata dentro la blockchain. La compagnia di assicurazione non potrebbe porre tanti ostacoli: al verificarsi dell'evento, se questo è attestato in modo crittografico, il beneficiario riceve la somma senza indugi. In un certo senso **il denaro non**

viene spedito, ma piuttosto il denaro si autospedisce al verificarsi di certe condizioni.

Esistono già delle società che per esempio lo fanno nel caso dei ritardi aerei e nel caso delle franchigie. Una società che si chiama 'Teambrella' copre piccoli danni attraverso uno smart contract che definisce una società di mutuo soccorso fra i membri. I membri possono votare per usare il fondo allocato nel contratto per risarcire o meno il danno richiesto da uno dei participanti.

Si possono dunque creare delle organizzazioni che potremmo definire di

diritto crittografico che nascono e muoiono nella blockchain, per distinguerle da quelle di diritto privato che necessitano di un contratto cartaceo in linguaggio legale ed un passaggio dal notaio.

La notarizzazione liquida o crittografica è proprio la caratteristica implicita della blockchain. In futuro in una pubblica amministrazione il protocollo e l'albo pretorio potranno essere implementati con questa tecnologia e godere immediatamente delle proprietà innate della blockchain che sono la trasparenza e l'immutabilità.

La trasparenza è tuttavia anche un

aspetto controverso. In alcuni casi è desiderabile e in altri rappresenta un problema. Bitcoin non garantisce l'anonimato anche se le transazioni non riportano nomi e cognomi ma solo degli pseudonimi. Un pagamento in bitcoin non è come un bonifico che viene validato da un'autorità centrale come una banca. I pagamenti con i bitcoin vengono invece validati da tutti e da nessuno in particolare. Tutti i partecipanti possono scaricare una copia della blockchain e analizzarla. Sono state sviluppate delle tecniche di intelligence per associare ad un indirizzo bitcoin la probabilità che questo appartenga ad una certa persona, o che sia riconducibile ad uno scambio

illegale.

Altre blockchain, come ad esempio Zcash, utilizzano delle tecniche crittografiche più avanzate che si chiamano zero knowledge proof, in cui non è più possibile fare nessun tipo di intelligence e la privacy è totale. Quindi l'informazione di chi paga, quanto paga e chi riceve diventa assolutamente indecifrabile. Questa è una tecnologia pensata per rendere il "bitcoin" completamente anonimizzato e non tracciabile.

La blockchain

Chiavi crittografiche, indirizzi e firme digitali

L'invenzione delle criptovalute **non viene dalla luna**. E' basata su concetti saldi e ben noti da decine di anni ormai. Di fondamentale importanza sono i concetti di crittografia asimmetrica e di firma digitale.

La firma digitale si ottiene grazie a due chiavi crittografiche, la chiave pubblica e la chiave privata (in pratica due lunghe sequenze di bit), che servono rispettivamente per ricevere e spendere i bitcoin. Vengono generate insieme, quella pubblica può essere diffusa pubblicamente, mentre quella privata deve essere custodita gelosamente. La loro principale caratteristica è che **ciò che viene cifrato con la chiave pubblica può essere decifrato solo con la chiave privata e viceversa**, un dato cifrato con la chiave privata può essere decifrato solo con la chiave pubblica.

Come equivalente nel mondo fisico,

possiamo immaginarci una scatola dotata di un lucchetto speciale e le due chiavi, la privata la teniamo unica, mentre la pubblica la duplichiamo e la diamo ai nostri amici. Se riempio la scatola con un certo contenuto e la chiudo con la chiave privata, chiunque trovando la scatola chiusa potrà aprirla ed essere sicuro che ciò che trova dentro è il mio contenuto originale, inalterato. Questo garantisce l'autenticità.
Viceversa chiunque potrà prendere la scatola vuota, metterci qualcosa dentro, chiuderla con la mia chiave pubblica e lasciarla da qualche parte. Solo io che possiedo la chiave privata potrò aprirla. Questo garantisce la sicurezza e la privatezza della consegna.

Esempio di chiave privata in cifre esadecimali

```
91149ee24f1ee9a6f42c3dd64c22877
```

Esempio di chiave pubblica in cifre esadecimali

```
042c6b7e6da7633c8f226891cc7fa8e
```

Esempio di indirizzo Bitcoin

```
13mtgVARiB1HiRyCHnKTi6rEwyje
```

In generale un indirizzo Bitcoin si

costruisce a partire dalla chiave pubblica e la chiave pubblica a partire dalla chiave privata. Ma allora perché non utilizzare direttamente la chiave pubblica come identificativo del destinatario di un pagamento? Perché è necessario o utile utilizzare l'indirizzo? Ci sono varie ragioni: intanto la chiave pubblica è molto lunga e scomoda da condividere con qualcuno. E' senz'altro più pratico trasmettere un indirizzo che ha una forma molto più compatta. In secondo luogo esistono indirizzi speciali detti multisig i cui coin possono essere incassati solo se si possiedono più chiavi private. Questo è particolarmente utile per costruire alcuni schemi di pagamento con firma congiunta. Ultima

ragione poi se si usassero le chiavi pubbliche invece degli indirizzi, un eventuale errore di battitura anche di un solo carattere su una chiave pubblica non potrebbe essere individuato ed il coin verrebbe semplicemente spedito nel nulla dato che nessuno avrebbe la corrispondente chiave privata. Al contrario un errore di battitura su un indirizzo sarebbe probabilmente innocuo. Per la precisione c'è un algoritmo che verifica che le cifre e le lettere che compongono un indirizzo Bitcoin rispettino delle specifiche regole. In tal modo i nodi semplicemente scartano le transazioni con un indirizzo che contiene un errore di battitura. Esiste tuttavia ancora una probabilità

pari a uno su quattromiliardi circa di commettere un errore che genera un indirizzo ancora sintatticamente corretto.

L'applicazione più interessante della chiave pubblica/privata è la cosiddetta **firma digitale**. Questa serve appunto per firmare i pagamenti in bitcoin (e tante altre cose). E' l'equivalente della firma fisica nel mondo digitale ma molto più sicura. Senza entrare nei dettagli operativi, con la nostra chiave privata possiamo generare una firma digitale per un certo documento, e chiunque riceve il documento con la firma potrà verificare grazie alla nostra chiave pubblica che noi e nessun altro abbiamo firmato e trasmesso quel documento.

Non doppiospendere

Chiarito che possiamo usare la firma digitale per attestare la paternità di un dato, di un documento o di una semplice affermazione in forma digitale, possiamo considerare un pagamento in bitcoin come la girata di un assegno.
Supponiamo che Alice abbia ricevuto un coin e voglia “girarlo” a Bob.

Crittograficamente significa che Alice userà la sua chiave privata per "firmare" digitalmente una proposizione tipo "*pagare 1 coin a Bob*" dove in realtà non comparirà il nome di Bob ma la sua chiave pubblica. In altre parole Alice usa la chiave privata per firmare la frase

"*pagare 1 coin alla chiave pubblica di Bob*" — firmato Alice

Possiamo renderci conto che non è necessaria nessuna blockchain per firmare questa "frase" ed in teoria Alice potrebbe spedire un coin a Bob attraverso il semplice uso di firme digitali. Tuttavia Alice protrebbe

spendere due o più volte il suo coin per esempio

Alice firma la frase “pagare 1 coin a Bob”
Alice firma la frase “pagare 1 coin a Bridget”

Senza un’autorità centrale che controlla, dato che Bridget e Bob in generale non sanno dell’esistenza l’uno dell’altro, Alice potrebbe barare doppio-spendendo il suo coin. **Ecco a cosa serve la blockchain**. In assenza di un’autorità centrale il protocollo prevede che tutti tengano una copia di tutte le transazioni. Tutti possono così verificare che Alice ha un solo coin.

Tutti possono capire che la seconda transazione in ordine di arrivo è illegale e la rifiutano. In questo modo tutte le copie della blockchain registreranno solo il primo pagamento, quello da Alice a Bob.

La scarsità di un oggetto digitale

Grazie alla blockchain nasce la prima volta il concetto di scarsità di un oggetto digitale. Siamo infatti abituati all'idea che una volta che abbiamo un file o un contenuto questo lo possiamo copiare e trasmettere in rete creando tutte le copie che vogliamo.

Anche nei casi in cui tali oggetti digitali siano protetti da qualche sistema di digital right management non è difficile superare queste limitazioni e riuscire a copiare il contenuto. Ad esempio un servizio come Netflix ci consente la visione ma non la copia, tuttavia un hacker ben equipaggiato potrebbe ottenere una copia del contenuto dai buffer in memoria. Alle brutte basterebbe riacquisire le immagini a schermo grazie ad una telecamera esterna.

Questa falsificazione e riproduzione di un oggetto digitale non può invece avvenire per i bitcoin. Infatti

occorrerebbe falsificare la blockchain ma questo è appunto ciò per cui la blockchain è una geniale invenzione: **ogni copia falsificata perde ogni credibilità e la rete la ripudia.**

Abbiamo visto che il protocollo di gestione della blockchain consente di creare e trasferire degli oggetti digitali, cioè i coin e di garantirne la loro scarsità, cioè la loro non falsificabilità. Ma la blockchain è un registro che deve essere mantenuto identico in migliaia di copie gestite da migliaia di diversi partecipanti non coordinati fra loro e potenzialmente malevoli. Com'è possibile che tale blockchain risulti integra e identica. Com'è possibile che

due o più partecipanti non scrivano in modo incoerente fra loro causando il proliferare di numerose copie non identiche.

La risposta è che scrivere sulla blockchain richiede di esibire una **prova di lavoro che è difficile da costruire ma molto facile da verificare**. In altre parole, nel caso di Bitcoin solo un partecipante ogni circa dieci minuti riesce a generare una prova di lavoro. Ma la prova di lavoro che cos'è?

Gioco: prova di lavoro con Sudoku

Il gioco crittografico alla base della blockchain richiede qualche nozione di crittografia per essere compreso. Invece di concentrarci sui dettagli matematici e crittografici cerchiamo di capire la logica della prova di lavoro in modo semplificato con un esempio basato sul

Sudoku. Pensiamo ad una maestra di scuola che affida a ognuno dei suoi allievi una copia del registro di classe contenente i voti. A seguito di una verifica assegna un voto ad uno di loro (ad esempio un 6 a Rossi) e chiede a tutti di compilare diligentemente la loro copia del registro inserendo il voto di Rossi. Rossi potrebbe essere tentato di assegnarsi un 10, mentre magari Bianchi che è un burlone, decide di mettere a Rossi un 4 e così via. A un certo punto non sarebbe più possibile capire quale copia del registro riporta fedelmente i voti. Un modo per evitarlo sarebbe appunto di scrivere insieme al voto anche la risoluzione di un gioco come il Sudoku, in tale Sudoku, una volta finito,

dovrà risultare nel primo angolo lo stesso numero che compare nel Sudoku del voto precedente (chiamiamo questo numero la firma), nel secondo angolo il voto assegnato allo studente (in questo caso il 6 dato a Rossi). Il terzo angolo risultante (quello in basso a destra) sarà la firma del nuovo Sudoku. Ora se uno volesse falsificare un voto dovrebbe ricominciare da capo a compilare il registro risolvendo tutti i Sudoku successivi, la cosa non è impossibile ma diciamo impraticabile se si ha poco tempo a disposizione ed in ogni caso si sa che gli studenti durante la ricreazione preferiscono fare altro.

Nota: hashing e SHA256

Con il termine SHA si indica una famiglia di cinque diverse funzioni crittografiche di hash sviluppate a partire dal 1993 per conto del governo americano. La sigla SHA sta per Secure Hash Algorithm.

Come ogni algoritmo di hash, l'SHA

produce un message digest, o "impronta del messaggio", di lunghezza fissa partendo da un messaggio di lunghezza variabile.

La prova di lavoro crittografica

La prova di lavoro non nasce con Bitcoin. E' stata inventata diversi anni prima per la prevenzione dello spam dal ricercatore Adam Beck che l'ha chiamata Hashcash.

Lo spamming è quella fastidiosa pratica di marketing in cui uno spammer spedisce messaggi non richiesti a migliaia o milioni di indirizzi email raccolti attraverso mezzi più o meno illeciti.

Questa pratica è molto diffusa per una ragione molto semplice: **spedire messaggi email è un'attività sostanzialmente a costo zero**. Non sarebbe una pratica così diffusa se per spedire ogni messaggio fosse richiesto il pagamento di una piccola tassa, l'equivalente di un francobollo digitale.

Chi ha inventato la posta elettronica, però, non aveva forse previsto l'avvento

dello spamming e non ha certo pensato al francobollo digitale.

L'inventore di Hashcash ha pensato di modificare il protocollo della posta elettronica con la cosiddetta prova di lavoro. In cosa consiste? In pratica abbiamo bisogno di una funzione matematica un po' speciale che si chiama hashing. Questa funzione ha una caratteristica particolare, prende come input un qualsiasi dato di lunghezza arbitraria e restituisce in uscita un numero in pratica molto grande compreso tra 0 e 99999…99999 con un numero prefissato di cifre, ad esempio 50 (è un esempio, potrebbero essere in numero diverso).

Un’altra caratteristica è che è molto difficile, o meglio impraticabile riuscire a capire quale dato in input abbia generato un certo valore di hash in output. Inoltre una piccola variazione dell’input provoca immense variazioni dell’output come nell’esempio sotto.

Hashing(“Ciao.”) -> 10342340234298987 9…23423456

Hashing(“Ciao!”) -> 95464546837774887 3…44453345

Se un messaggio di posta elettronica “tradizionale” lo possiamo immaginare così:

Sender=spammer@cattivo.com
Recipient=mario@foo.com
Body="Contenuto del messaggio e allegati vanno qui"

Lo spammer genera tanti messaggi tutti identici in cui cambia solo il recipient, ovvero il destinatario.

Nella versione del messaggio modificata con hashcash, questo diventa

Sender=spammer@cattivo.com
Recipient=mario@foo.com
Hash=000045…456345
Nonce=un numero generato a caso
Body="Contenuto del messaggio e

allegati vanno qui”

Osserviamo che una caratteristica del campo Hash è quella di iniziare con un certo numero di zeri, in questo esempio quattro. Questo potrebbe essere il nostro target. Ovvero il server di posta elettronica in ingresso accetta solo messaggi con Hash che iniziano con un certo numero di zeri. Lo spammer per generare l’Hash deve concatenare insieme Recipient, Nonce e Body e vedere se ottiene un Hash con quattro zeri iniziali.

Hashing(Recipient ... Nonce ... Body) inizia con quattro zeri?

Se non ci riesce può cambiare il Nonce e riprovare. Dopo un certo numero di tentativi troverà sicuramente un Hash obiettivo. Tuttavia la cosa non è immediata, in funzione di quanti zeri abbiamo come obiettivo potrebbe richiedere qualche secondo o qualche minuto.

Per il server ricevente invece la verifica che il campo Hash corrisponda al risultato di Hashing(Recipient … Nonce … Body) e che questo risultato inizi con quattro zeri è una cosa immediata.

Creazione onerosa — Verifica immediata

Questo implica che lo spammer per spedire un messaggio ha bisogno di tempo, per esempio in media 30 secondi. Mentre chi riceve fa uno sforzo minimo per verificare. Ora uno spammer che volesse spedire 12,000 messaggi avrebbe bisogno di 6000 minuti ovvero 100 ore. Evidentemente questo renderebbe lo spamming molto meno appetibile.

Lo “sforzo” di calcolo è invece ancora accettabile per un utente normale che spedisce qualche decina di messaggi al giorno. Questo algoritmo ha ispirato Satoshi Nakamoto nella progettazione dell’algoritmo proof-of-work che viene

usato per inserire e confermare le transazioni nella rete Bitcoin.

Blockchain: take-aways

I pagamenti in Bitcoin sono come delle girate di assegni ma firmati digitalmente.

La firma digitale si ottiene a partire da tre elementi fondamentali, la chiave pubblica, la chiave privata e una

funzione non invertibile detta di hashing.

Bitcoin è la prima soluzione che risolve il problema di registrare le transazioni tra le parti senza che nessuno abbia più autorità degli altri.

La blockchain non consente a nessuno di doppio spendere e nello stesso tempo non esiste un singolo punto di vulnerabilità poiché nessun nodo è essenziale al funzionamento.

La catena dei blocchi di Bitcoin garantisce l'immutabilità delle transazioni e la loro integrità anche in presenza di molti utenti malevoli purché non coordinati fra loro e che accedono

alla rete senza autenticazione.

Gli smart contract

Introduzione

In questo capitolo ci sono alcuni dettagli tecnici e anche alcuni pezzi di codice. E' proprio necessaria quest'immersione nella tecnica? In fondo potremmo parlare della criptoeconomia senza sporcarci le mani con il codice sorgente. La verità è che mai come nel caso delle criptovalute e degli smart contract

l'aspetto economico e l'aspetto tecnologico sono così legati. Nascono come gemelli non separabili alla nascita. **Un errore in un'istruzione di un programma qui significa denaro che sparisce per sempre dalla rete.**

Detto questo, anche se non sei un programmatore **ti invito a leggere il capitolo e non preoccuparti se non capisci ogni dettaglio**.

La comprensione anche parziale di alcuni meccanismi di Bitcoin o di Ethereum è importante per tutti. Se venti anni fa ti avessero parlato di numero IP, URL, indirizzo email forse sarebbe stato duro capire, eppure molti di questi elementi ormai ci sono più familiari e chi li ha capiti per primo ha avuto

qualche vantaggio.

Se invece sei un programmatore allora considera questa come una breve e rapida introduzione all'argomento. Imparare a programmare gli smart contract e creare applicazioni decentralizzate con Ethereum , richiederebbe molto più spazio e tempo di quello disponibile per questo testo. Dopo la lettura di questo capitolo ti invito a consultare la documentazione ufficiale del linguaggio solidity e ordinare in prevendita il libro *Mastering Ethereum* di Antonoupulos & Wood la cui uscita è prevista per Febbraio 2018.

Bitcoin e i contratti

Prima di avventurarci nel mondo degli smart contract Ethereum facciamo una rapida panoramica di quello che comunque è possibile già oggi con la rete Bitcoin.

Script

Finora abbiamo espresso tutti i nostri ragionamenti sul funzionamento della blockchain con un linguaggio naturale e non ci siamo occupati di come questa logica sia poi veramente implementata nelle macchine dei nodi Bitcoin.

E' utile precisare che quando noi costruiamo un semplice pagamento tipo

paga 1 coin a Bob — firmato Alice

In realtà avviene qualcosa di abbastanza complesso. Una transazione Bitcoin come quella descritta si aggancia a quello che chiamiamo

unspent transaction output (UTXO), o più brevemente output di una transazione precedente che in questo caso mette a disposizione 1 coin. Per essere più precisi andrebbe rivista in questi termini:

Prendi come input l'output della transazione A e crea un nuovo output di 1 coin a favore di Bob — transazione B firmata da Alice

Di fatto ogni input ed ogni output sono dei veri e propri programmi scritti in un linguaggio di programmazione chiamato Script. Il programma sull'output viene chiamato **locking script**. Questo significa che Alice ha

bloccato un coin attraverso un locking script e chiunque potrebbe in teoria fornire un **unlocking script**, ma in pratica solo Bob grazie alla sua chiave privata potrà farlo.

In pratica locking script e unlocking script vengono messi in esecuzione uno dopo l'altro e se il risultato complessivo dell'esecuzione è true (vero booleano) allora il coin viene speso, altrimenti no.

Storicamente il locking script è stato chiamato con il codename scriptPubKey mentre l'unlocking script è stato chiamato scriptSig. Questi sono i campi che trovereste in una transazione Bitcoin se andaste a ispezionarne il contenuto.

Un pagamento verso un indirizzo ad esempio è codificato normalmente con uno script di tipo pay-to-pubkey-hash dentro un output con i seguenti operation codes (tratto da https://en.bitcoin.it)

scriptPubKey:
OP_DUP
OP_HASH160
<pubKeyHash>
OP_EQUALVERIFY
OP_CHECKSIG

scriptSig:
<sig>
<pubKey>

Nella tabella sotto viene mostrata l'esecuzione passo passo dello script che risulta dall'unione dell'unlocking script con il locking script (tratto da https://en.bitcoin.it)

Script	Stack
<sig> **<pubKey>** **OP_DUP** **OP_HASH160** **<pubKeyHash>** **OP_EQUALVERIFY** **OP_CHECKSIG**	
OP_DUP **OP_HASH160**	

<pubKeyHash> OP_EQUALVERIFY OP_CHECKSIG	**<sig> <pubKey>**
OP_HASH160 <pubKeyHash> OP_EQUALVERIFY OP_CHECKSIG	**<sig> <PubKey> <PubKey>**
<pubKeyHash> OP_EQUALVERIFY OP_CHECKSIG	**<sig> <PubKey> <PubKeyH**
OP_EQUALVERIFY OP_CHECKSIG	**<sig> <PubKey> <PubKeyH <PubKeyH**
OP_CHECKSIG	**<sig> <PubKey>**

Sebbene relativamente semplici, la leggibilità degli script Bitcoin non è il massimo. Per questo motivo nelle sezioni successive non farò uso degli Operation Codes per descrivere la logica dei contratti ma continuerò ad utilizzare pseudo codice e descrizioni in lingua italiana per garantire una migliore leggibilità.

Schema multisig

Una cosa importante da chiarire è come il concetto di contratto in questo contesto non è esattamente quello del contratto così come definito nel codice civile. Diciamo per semplicità che uno smart contract su blockchain non è né smart né contract e che se proprio vogliamo definirlo lo possiamo pensare come una forma di "denaro programmabile".

Una prima tipologia di contratto è quella dello schema multifirma o multisig per semplicità. Questo è un tipo di pagamento più articolato del semplice Alice paga 1 coin a Bob. La logica del

pagamento potrebbe essere descritta invece con qualcosa tipo:

*Alice **paga** 1 coin che **Bob** può incassare **se** Trent è d'accordo.*

In questo esempio abbiamo la logica di un pagamento con veto o approvazione di un terzo. Questa tipologia consente di implementare gli escrow, ovvero degli arbitri che decidono sulla regolarità di una transazione. Supponiamo che Alice acquisti una bicicletta online da Bob e paghi con i coin. Nel caso di acquisto diretto potrebbero verificarsi diverse situazioni.

Alice paga Bob ma Bob non spedisce la bicicletta o la bicicletta è rotta.

Bob spedisce la bicicletta ma Alice non paga

Alice paga Bob e Bob spedisce una bicicletta in buone condizioni

Solo l'ultimo caso è quello onesto, negli altri qualcosa va storto. Ricordiamo che un pagamento in criptovaluta è irreversibile e normalmente non conosciamo l'identità di chi sta dall'altra parte rendendo vano ogni tentativo di recupero.

Spesso nella nostra terminologia siamo abituati a parlare di transazione

riferendoci al solo pagamento. In realtà la transazione è bidirezionale, da una parte c'è un pagamento e dall'altra un trasferimento di un bene o servizio che può essere virtuale o fisico. Nel caso di un oggetto fisico, come ad esempio un immobile, la transazione va conclusa grazie all'intervento di un qualche intermediario. Purtroppo questo manda all'aria una buona parte dell'ideale disintermediazione che vorremmo dalla blockchain.

Nel caso particolare della bicicletta possiamo costruire un contratto di questo tipo, consideriamo un coin bloccato da Alice che può essere sbloccato solo da 2 su 3 firme fra quelle

di Alice, Bob e Trent. A questo punto possiamo analizzare i vari casi precedenti.

Alice spedisce il coin 2 su 3 a Bob ma Bob non spedisce la bicicletta. Bob però non può da solo sbloccare il coin a proprio favore. Alice si lamenta con Trent, Trent non firmerà lo sblocco del coin a meno che Bob non provi di aver veramente spedito la bicicletta.

Alice spedisce il coin 2 su 3 a Bob ma la bici e rotta. Alice si lamenta con Trent e documenta attraverso immagini che la bicicletta è rotta. Bob non è in grado di provare che fosse integra quindi Trent non sblocca il coin.

Bob spedisce la bicicletta ma Alice non vuole sbloccare il coin lo stesso. Bob si rivolge a Trent e le loro due firme possono ora sbloccare il coin a favore di Bob.

Alice riceve la bicicletta ed è soddisfatta. Alice e Bob firmano per sbloccare il coin a favore di Bob. Trent non deve intervenire.

https://bitcoin.stackexchange.com/que
will-multisig-addresses-work

Pagamenti a tempo in Bitcoin

Un'altra caratteristica delle transazioni Bitcoin è che esse possono specificare un tempo prima del quale non sono da considerarsi valide. In altre parole possiamo ottenere lo stesso effetto di un assegno post datato. Questo, se andiamo in banca ad incassarlo, verrà rifiutato dal cassiere. Allo stesso modo esiste in Bitcoin un attributo della transazione che si chiama nLockTime, che si esprime in blocchi, e rappresenta il numero di blocco prima del quale la transazione non deve essere accettata dalla rete.

Oltre il già citato e implementato nLockTime, Bitcoin si doterà presto di

una nuova modalità di pagamento temporizzato detta Hashed Timelock Contracts. In pratica chi riceve un pagamento deve accettarlo entro una certa data, mostrando un segreto o la prova di possedere un segreto, o altrimenti il pagamento viene revocato. Questo apre numerose possibilità di pagamenti condizionali. Una di queste possibilità è quella di transazioni atomiche di scambio fra due blockchain diverse dette atomic swaps. In pratica potrà essere realizzata lo scambio tra ad esempio bitcoin e litecoin (altre coppie) direttamente tra utenti senza passare da un exchange ed in maniera atomica, ovvero le due transazioni: *Alice riceve M bitcoin da Bob* e *Bob riceve N*

litecoin da Alice avvengono insieme oppure nessuna delle due avviene.

Channel per micropagamenti

Abbiamo visto come la proof-of-work, ossìa il “gioco” che consente agli utenti di generare nuovi bitcoin e alla rete di mantenere l’integrità della Blockchain, richiede molti calcoli e molti calcoli significa molta energia. Questo è un dato di fatto. Ogni volta che un microprocessore esegue una somma o una sottrazione, un briciolo di energia viene dissipato in forma di calore. Un altro dato di fatto è che i calcoli necessari per la proof-of-work non forniscono, a parte la sicurezza delle transazioni, nulla di veramente interessante o utile per l’umanità come ad esempio la scoperta di un nuovo

numero primo o di un segnale di vita alieno nel cosmo.

Ma quanto costa Bitcoin in termini energetici? Una stima da me effettuata nel 2016 e confermata da altre fonti rivelava che l'energia spesa per ogni transazione era circa 19KWh, non ci sono ragioni per ritenere che sia diminuita in modo sostanziale. Ci tengo a precisare che il nostro caffè a noi costa sempre e solo come un caffè anche se lo paghiamo in bitcoin. Questo costo energetico non intacca il nostro conto. E' un costo di cui si fa carico la rete nel suo complesso e che per ora viene ritenuto sostenibile dai partecipanti perché questi riescono a generare nuovi bitcoin e i bitcoin valgono molti dollari.

Per questa ragione possiamo dire che Bitcoin è un'eccellente rete per transare grosse somme ma non molto efficiente per le piccole transazioni. A questo scopo è nata la proposta di spostare fuori dalla blockchain i micropagamenti. Il concetto è molto semplice ed è ispirato da ciò che avviene già nella realtà. Pensiamo al salumiere che ci fa credito, compriamo qualcosa e diciamo - metti in conto. Ecco questa è in sostanza una micro transazione che non avviene nel circuito ufficiale, non c'è uno scambio di denaro né in contanti né tantomeno attraverso il sistema bancario. Eppure la transazione è

avvenuta. Solo alla fine del mese il conto ci viene presentato e in quel momento salderemo al salumiere tutte le sue spettanze annotate nel suo registro locale. In altre parole abbiamo trasformato decine di piccole transazioni in un unica transazione ufficiale che prevede il trasferimento di moneta.

Naturalmente nel caso di Bitcoin le cose si complicano perché devono poter avvenire a distanza e noi non godiamo della fiducia personale del salumiere che ci fa credito. Tuttavia ispirati dal meccanismo appena descritto possiamo concepire un modo per ottenere un risultato simile, ovvero quello di lasciare che Alice e Bob si sbrighino da

soli tutta una serie di piccole transazioni off chain, ovvero che non arrivano mai alla blockchain, e poi una transazione finale che invece sarà validata e trasmessa alla blockchain.

Vediamo un esempio. Alice è la casalinga e Bob il salumiere, o se preferite invertiamo le parti. Supponiamo che Alice voglia fare molti acquisti nella stessa giornata, a tale scopo crea una transazione T di tipo multisig 2 su 2 con un output di 100 coin che solo Bob e Alice insieme possono sbloccare.

Alice firma una seconda transazione R postdatata alle ore 24 che ha come

input l'output di 100 coin della precedente transazione T e due output, uno di 100 a favore di Alice e uno di 0 a favore di Bob. Alice chiede a Bob di firmarla a sua volta. In questo modo la transazione R se inviata alla blockchain immediatamente viene rifiutata.

Se nel corso delle 24 ore Alice non acquista nulla, la transazione R a un certo punto diventerà valida e Alice potrà incassare i 100 coin che aveva momentaneamente bloccato.

Se invece Alice dopo 1 ora decide di acquistare beni del valore di 10 coin, allora firmerà una nuova transazione R', questa volta post datata alle ore 23

(prima della transazione R), e con i nuovi output 90 per Alice e 10 per Bob. La transazione R' sarà valida **più presto** della transazione R, questo garantisce a Bob di incassare i 10 coin se Alice non farà altri acquisti. Alice potrebbe decidere invece di fare altri acquisti per esempio del valore di 25 coin. A questo punto dovrà creare una nuova transazione R'' che come input i 100 coin di T, sarà postdatata alle ore 22 (prima della R') e distribuirà 35 coin a Bob e 65 ad Alice.

Questo meccanismo prevede che ogni nuova transazione sia gestita da Alice e Bob senza passare dalla blockchain. Per evitare che possa essere

inviata un saldo non aggiornato ogni nuova transazione prevede una postdatazione più “imminente” delle precedenti.

In questo modo, quando il momento per l’ultima transazione è ormai arrivato questa viene spedita in blockchain e in un unica transazione di saldo vengono distribuite le spettanze a tutti i partecipanti.

In questo modo Alice salda il conto del salumiere Bob.

Lightning Networks

I canali di micropagamento sono una soluzione ideale per risolvere offchain le piccole transazioni e utilizzano la blockchain solo per il saldo finale. Lo svantaggio è che però richiedono un coordinamento specifico fra due partecipanti per mettere in piedi questo schema. E' possibile tuttavia costruire delle vere e proprie reti in cui gli archi sono dei canali di micro-pagamento. Ad esempio se Alice ha un canale aperto con Bob e Bob un canale aperto con Charlie, sarebbe possibile per Alice spedire micro-pagamenti a Charlie. Questa tipologia di connessione fra diversi canali di micro-pagamenti

consente la realizzazione di vaste reti di transazioni off-chain. Queste reti sono dette Lightning Networks.

La caratteristica di queste reti è che consentiranno di scalare il numero di transazioni Bitcoin in modo massivo eguagliando se non superando il cosiddetto throughput dei circuiti di pagamento tradizionali come Visa e Mastercard, ma nello stesso tempo garantendo una maggiore sicurezza delle transazioni.

Quando qualcuno obietterà come al solito che Bitcoin non scala con il numero di transazioni potremmo rispondere che “La blockchain non scala

ma Bitcoin scala benissimo”.

Ethereum e i contratti

Sebbene questo libro non sia un manuale di Ethereum dobbiamo introdurre alcuni concetti chiave per poter poi parlare di criptoeconomia, di smart contract e di ICO.

Breve storia di Ethereum

Ho già parlato di Ethereum, ma forse non ho raccontato abbastanza della sua storia e del suo inventore.

Vitalik Buterin è un ragazzo di 20 anni, testa un po' quadrata, magrissimo, probabilmente un genio. Lui e gli altri coinventori di Ethereum hanno pensato "perchè limitarsi ad una moneta per computer? Perché non mettere in piedi un intero universo in cui i computer rispettano dei contratti digitali molto più articolati. Perché invece di un denaro digitale non creare un denaro programmabile?"

Vitalik ha in mente il computer

definitivo che non può essere mai spento e dove programmi agiscono, scambiano transazioni, pagano, rispettano leggi. Quali leggi? Quelle scritte nei contratti che non possono essere infranti senza conseguenze. **Code is law**. Ricordiamocelo perché dovremmo tornare su questo.

Vitalik e soci attraverso una società chiamata EthSwiss, mettono in piedi uno dei più importanti crowdfunding della storia ma senza passare da Kickstarter. EthSwiss vende token del nascente progetto - paghi in bitcoin ottieni in cambio ether. Nasce Ethereum ed in un certo senso nasce anche il fenomeno ICO (vedi sezione apposita).

All’inizio non sono state rose e fiori. Ecco le critiche più comuni (riportate sotto come trovate sui forum):

They are asking an insane 30k BTC
50% pre mined
They are setting a price out of nothing for 1 eth, while it’s not even out yet
“ex” Goldman Sachs pricks are supporting this project

A differenza di Bitcoin, Ethereum suona meno anarchico, più politicamente corretto e forse piace di più alle istituzioni e alle corporation. Microsoft

aggiunge nel suo celebre Visual Studio uno strumento per programmare in solidity, il linguaggio per gli smart contract di Ethereum e IBM si lancia in un progetto che si chiama Adept che in un colpo solo incrocia due keyword molto in voga: **Internet of things e blockchain**.

Il 2016 è un bel momento di hype per la blockchain. Tutti parlano di blockchain, ci sono blockchain per tutti i gusti …permissioned, unpermissioned, privata, pubblica, blockchain per i cibi, una per le piante, una per i dati, una per i diritti d'autore, blockchain veloci, blockchain senza blocchi, blockchain senza chain, database cammuffati

da blockchain etc.

Ragazzini di vent'anni salgono in cattedra a raccontare il futuro e ai banchi i vecchi professori prendono appunti.

Parliamo degli smart contract. Questi sono in realtà dei programmi che vengono messi in funzione sulla blockchain e vengono eseguiti in maniera identica in tutti i nodi della rete. Ogni programma lo possiamo immaginare come una macchina a stati, o meglio ogni programma ha dei registri in memoria che chiamiamo variabili di stato. Quando il programma riceve una transazione, questo cambia il suo stato. Questi stati non sono memorizzati nella

memoria di un solo computer ma in quella di tutti i computer (della rete Ethereum). In pratica le sue variabili sono mantenute nella blockchain alla stessa stregua del saldo di un conto. Un computer terribilmente inefficiente perché in pratica un programma fatto di poche istruzioni viene eseguito da migliaia di computer in modo identico per far funzionare il protocollo di consenso e dire: **acconsento, siamo d'accordo sullo stato successivo del programma**.

Terribilmente **inefficiente ma praticamente indistruttibile**, una volta che parte non può essere spento se spegnersi a comando non è previsto nel

suo codice sorgente.

Ethereum Virtual Machine

Abbiamo visto che la logica dei contratti eseguibili con Bitcoin ed il suo linguaggio Script ha il limite di non essere iterativa, in pratica non è possibile prevedere una qualche azione che all'interno di una transazione Bitcoin si ripeta con la logica

INIZIO:
fa qualcosa,
…,
TORNA all'INIZIO.

Questa tipologia di iterazione detta loop è tipica dei linguaggi di

programmazione. Ma il design Bitcoin esclude questa tipologia di iterazioni per ragioni di sicurezza. Una logica malevola del tipo

INIZIO:
TORNA ALL'INIZIO

lanciata sulla blockchain sarebbe un disastro. Migliaia di nodi obbedirebbero ai comandi e non uscirebbero mai dal loop, la rete sarebbe impegnata per sempre.

Ethereum invece non esclude questa possibilità e propone una soluzione basata sul concetto di GAS (vedi sezione apposita). Ethereum si può

considerare un'implementazione di blockchain ispirata a Bitcoin ma non derivata direttamente dal suo codice sorgente come ad esempio Litecoin. Ethereum, come Bitcoin, usa il Nakamoto consensus, ovvero la prova di lavoro che abbiamo descritto nel capitolo dedicato a Bitcoin. La comunità Ethereum è tuttavia fermamente orientata a modificare il protocollo di consenso passando ad un sistema cosiddetto proof-of-stake. Questo protocollo dovrebbe consumare molta meno energia e garantire un livello comunque accettabile di sicurezza. Non possiamo approfondire qui il funzionamento della proof-of-stake, se sei curioso ti invito a consultare questa risorsa

[https://it.wikipedia.org/wiki/Proof-of-stake].

Ogni nodo Ethereum implementa una Ethereum Virtual Machine, ovvero un modulo software in grado di emulare il comportamento di un pseudo-processore di comandi elementari che possono compiere operazioni aritmetiche, crittografiche, di controllo come IF e cicli FOR.

Non entriamo nei dettagli della EVM, diciamo semplicemente che è come una CPU ma virtuale e soprattutto dedicata all'esecuzione di un set di istruzioni per la blockchain e non un set di istruzioni general purpose come

quelle ad esempio eseguite dal tuo PC.

Sotto è riportato un esempio, leggibile, di istruzioni della EVM.

```
.code
  PUSH 60
  PUSH 40
  MSTORE
  PUSH 0
  DUP1
…
```

Non descrivo qui il significato dei vari Operation Codes. Chi volesse approfondire può dotarsi di pazienza e affrontare le pagine di documentazione online a partire dall'impegnativa lettura del yellow paper di Gavin Wood.

Come avviene per i linguaggi di programmazione per i comuni PC, anche per la EVM non è molto pratico scrivere un contratto direttamente utilizzando gli OP_CODES come nell'esempio. In genere quello che si fa è di utilizzare un linguaggio di programmazione detto di alto livello, che non significa extra lusso o alta qualità, significa semplicemente che se il basso livello è il livello della macchina, quello alto è il livello dell'uomo. Un linguaggio di alto livello è semplicemente più facile da leggere, scrivere e capire anche per un essere umano restando tuttavia un linguaggio formale. Tra questi linguaggi, quello che va per la maggiore su Ethereum è detto solidity ed un semplice pezzo di codice

potrebbe apparire così.

```
contract Hello{
    uint public x=0;
}
```

Il frammento è sostanzialmente **inutile** come contratto. Dichiara una variabile x intera positiva di valore pari a zero e non fa altro. Dato che la EVM capisce solo gli OP_CODES del suo linguaggio macchina, sarà necessario utilizzare un compilatore, ovvero un programma che trasforma solidity in Operation Codes.

Nota: Turing completo

La Turing completezza è la proprietà dei modelli di calcolo che hanno lo stesso potere computazionale di una macchina di Turing universale (MdTu).

Non sai cos'è una macchina di Turing? Non ti preoccupare.

I più comuni linguaggi di programmazione, sia imperativi che funzionali sono Turing completi. Script, il linguaggio delle transazioni Bitcoin NON è Turing completo. **Il linguaggio delle transazioni Ethereum è Turing completo**.

Attenzione: la completezza non indica una superiorità di Ethereum rispetto a Bitcoin. E' una scelta di design. Le scelte effettuate dalla comunità Bitcoin sembrano dettate dal principio di non permettere agli utenti di scrivere interi programmi eseguiti in blockchain perché questi aumentano la superficie d'attacco e rendono la blockchain complessivamente meno sicura. Chi ha progettato Ethereum ritiene che invece valga la pena correre il rischio per agevolare l'evoluzione della blockchain da mezzo di pagamento a macchina universale di calcolo decentralizzato. Due visioni opposte, entrambi rispettabili.

GAS

Ogni transazione Ethereum contiene dunque delle istruzioni elementari. L'idea è che ogni istruzione venga eseguita e contabilizzata con un'unità di costo macchina chiamata GAS, il GAS a sua volta ha un valore in ether che cambia nel tempo a seconda della dinamica di domanda e offerta. In questo modo ogni transazione deve fornire una dote di ether supplementari per la sua esecuzione. Un loop infinito malevolo si esaurirebbe semplicemente per fine del GAS.

Ogni istruzione ha un certo costo espresso in GAS. Ogni unità di GAS a

sua volta avrà un certo costo espresso in ether. Perché non è stato direttamente assegnato un costo in ether alle istruzioni? La ragione è che si è voluto separare il costo computazionale da quello economico. Se è accettabile che una somma fra interi costi l'equivalente di 0.00001$ non è altrettanto accettabile che a causa della volatilità del prezzo degli ether questa arrivi a costare 1$.

Quando creiamo una transazione, ovvero una chiamata ad una funzione di un contratto, possiamo decidere due parametri: uno è il **gas limit** e l'altro è il **gas price**. Il gas limit deve essere opportunamente stimato perché se mettiamo un gas limit troppo basso

rischiamo di finire il GAS nel mezzo della transazione e quindi non riusciamo a completarla., e per sua natura una transazione o è completa oppure viene ripristinato lo stato precedente. D'altro canto un GAS troppo abbondante non è un problema perché tutto il GAS che avanza ci viene restituito. Questo a patto che la transazione vada fino in fondo, se il miner si blocca per un errore il GAS viene perso.

Il GAS price invece è un altro parametro interessante. Il GAS ha un prezzo grosso modo equivalente fra i vari miner, il nostro wallet è in grado di indicarcelo e questo ci permette di non usare un GAS price fuori mercato. Se proponiamo un GAS price troppo basso

la nostra transazione potrebbe non essere presa in considerazione dai miner, viceversa un GAS price alto diventa un incentivo per far entrare immediatamente la nostra transazione nel prossimo blocco. Tuttavia in alcune situazioni come delle importanti crowdsale e ICO, per evitare che la competizione nell'acquisto di token possa aumentare il GAS price a dismisura e causare un problema a tutta la rete, i contract che vendono i token tendono a scartare i pagamenti con GAS price troppo alti.

Indirizzi Ethereum, account e contratti

Così come per Bitcoin, anche Ethereum considera che le transazioni avvengano tra diversi account. Ogni account è rappresentato da un indirizzo esadecimale unico. Per esempio:

0x8387A9f497353af8a54269824f227

Così come in Bitcoin, non dovete chiedere a nessuno il permesso per la creazione del vostro o dei vostri account Ethereum. Basta installare un wallet e creare i corrispondenti indirizzi, chiavi pubbliche e chiavi private.

Una caratteristica peculiare di Ethereum è che alcuni agenti, detti in modo infelice contratti, possono spedire e ricevere denaro esattamente come gli account umani e hanno di fatto un loro indirizzo Ethereum. Questa caratteristica apre a numerose possibilità che vedremo di seguito.

Wallet

Il wallet in genere è un software che consente di gestire il proprio salvadanaio cripto. Un wallet normalmente gestisce uno o più indirizzi Bitcoin/Ethereum o altra valuta e per ognuno di questi conserva la chiave privata senza la quale non sarebbe possibile spendere i relativi coin. Ci sono vari tipi di wallet, alcuni gestiscono le operazioni nel computer dell'utente mentre altri sono solo delle interfacce a dei servizi remoti. Questi ultimi sono in generale più semplici da usare e installare (per esempio sugli smartphone) ma attenzione, di fatto

dovete aver fiducia di chi li gestisce perché sono loro ad avere le chiavi private e quindi di fatto possono fare quello che vogliono con i vostri coin. E' come usare un servizio di banking ma con l'aggravante che non è regolamentato come le banche vere. I principali wallet per Ethereum sono:

Geth è il client Ethereum ufficiale, è un'applicazione a terminale e quindi ha un'interfaccia a riga di comando. Non è quello che la maggiorparte degli utenti utilizza direttamente, tuttavia è un elemento importante al quale generalmente è possibile collegare l'applicazione Mist che invece è un'applicazione dotata di interfacce

utente più facili da usare.

Mist è il wallet ufficiale che in pratica si collega al nodo geth. Usando Mist/Geth si deve sempre scaricare la blockchain sul proprio disco in quanto questi nodi sono dei veri e propri full node e possono anche occuparsi del mining, sebbene questa possibilità sia poco remunerativa su un normale PC.

Parity è prodotto dall'azienda omonima di Gavin Wood ex membro della Ethereum Foundation e di EthDev. Come Geth, anche Parity è un full node e richiede di scaricare tutta la blockchain. Il sync con la blockchain risulta tuttavia molto rapido. Parity è suddiviso in una

parte nodo vero e proprio ed un server web che consente di controllare tutte le operazioni grazie ad un'interfaccia HTML disponibile attraverso il browser all'indirizzo locale http://127.0.0.1:8180/

MyEtherWallet è un ottimo client che non richiede l'installazione sul nostro PC di nessun programma né tantomeno di scaricare la blockchain. Infatti MEW è un sito web, ma pur essendo un app web non richiede che noi apriamo un account presso i loro server o che loro gestiscano le nostre chiavi private e gli altri segreti crittografici. Tutto il software è dentro

la pagina web in forma di Javascript e HTML e tutte le operazioni crittografiche sono svolte nella pagina stessa, in locale. Questo è molto interessante e si presenta come una soluzione allo stesso tempo pratica, non installiamo nulla sul nostro PC, e sicura visto che le nostre chiavi non lasciano mai il nostro PC. Il codice è open source e visibile su github. Detto questo, un po' di ansia nel vedere le nostre chiavi dentro il browser resta. Controllate bene che siate collegati effettivamente al sito https://www.myetherwallet.com/ . Purtroppo ci sono degli scam wallet ottimi imitatori di tutta l'interfaccia utente dell'originale, e in questi se inserite la chiave privata vi rubano gli

ether.

Metamask è un extension del browser Chrome. Ha la simpatica caratteristica di poter funzionare come un bridge tra le applicazioni Ethereum che girano nel browser e la blockchain stessa, senza però la necessità di avere la blockchain scaricata in locale. Script e pagine HTML potranno usare Metamask (se installato) per generare transazioni e spedirle sulla blockchain. Metamask si presenta con un'ottima interfaccia grafica e ovviamente può essere usato come borsellino per spedire e ricevere ether.

Hello World in solidity

I nodi Ethereum alla fine non sono altro che dei processi in esecuzione su dei computer. Tali processi sono dotati di un componente detto EVM il quale è come un emulatore di un computer esso stesso.

Ma perché questa complicazione? Perché i nodi Ethereum devono emulare dei computer virtuali quando girano a loro volta su dei computer reali?

In generale questa è una buona soluzione per avere una macchina virtuale di Ethereum che non interferisce troppo con quella reale che la ospita, e in questo modo è possibile che la macchina virtuale abbia un set di

istruzioni elementari indipendenti dall'architettura fisica del computer reale.

Questo permette anche di scrivere i programmi per Ethereum con diversi linguaggi di programmazione anche molto diversi fra loro ma poi tutti sono compilati in codice binario che può essere essere compresi ed eseguiti dalla macchina. Nel caso di EVM questo è vero ed infatti esistono diversi linguaggi per scrivere i contratti che girano sulla blockchain. Tra questi linguaggi ce ne sono tre in particolare che hanno avuto un certo sviluppo Solidity, Serpent e LLL. Ma in pratica Solidity è quello maggiormente usato e aggiornato ed è l'unico di cui mostreremo qui qualche

breve esempio.

Degli altri due dirò soltanto che Serpent è una specie di Python (in natura avrebbe senso il contrario), mentre LLL è simile a Lisp. E se conoscete Lisp sapete che non è il caso di approfondire qui la sintassi.

A questo punto possiamo scrivere un primo contratto semplice semplice, un hello world che ci permette di prendere pratica con alcuni concetti.

```
contract Hello{
    function Hello(){}
    function () payable{}
}
```

Una funzione ha lo stesso nome del contract ed è usata come costruttore

mentre l'altra chiamata ha la modificatore payable che indica che una transazione che la invoca può anche portare con sé un pagamento che va a rimpinguare il saldo del contratto stesso che, ricordiamoci, può ricevere e spedire ether.

Come si può vedere le funzioni sono vuote. Volutamente in questo caso perché il nostro è un contratto minimale che di fatto non fa nulla a parte **esistere** ed eventualmente ricevere degli ether. Non essendoci poi delle funzioni che permettono di estrarre gli ether depositati di fatto questo contratto è un pozzo senza fondo dal quale non è possibile recuperare i fondi depositati. Quindi non serve a niente se non a

bruciare ether e a mostrare la struttura elementare di un contratto. Per una panoramica più completa della documentazione di solidity e sulla struttura dei contratti vi invito a leggere la doc ufficiale.

Una volta compilato lo possiamo lanciare su blockchain con deploy et voilà, il nostro contratto è in esecuzione pronto a ricevere pagamenti.

Il contratto compilato per la EVM assume un aspetto sicuramente molto più criptico, qualcosa tipo:

PUSH1 0x60
PUSH1 0x40
MSTORE
CALLVALUE
ISZERO

PUSH1 0xE

[…]

Come già detto, ogni contratto una volta creato e lanciato nella blockchain ha un suo indirizzo ed uno suo saldo esattamente come un account utente. E' possibile verificare entrambi attraverso etherscan.

Etherscan
The Ethereum Block Explorer

HOME

Contract Address 0xab7c74abC0C4d48d1bdad5DCB26153FC8780f83E

Contract Overview

ETH Balance:	1,500,000.001341970942807 89 Ether
ETH USD Value:	$451,335,000.40 (@ $300.89/ETH)
No Of Transactions:	203 txns

Un contratto più interessante

Il codice che abbiamo visto nella sezione precedente non fa veramente nulla. Vediamo un esempio più interessante. Sviluppiamo qui di seguito un contratto che riceve i pagamenti solo all'interno di un certo periodo e li rifiuti successivamente.

Per fare questo è necessario definire una speciale espressione di solidity detta **modifier.**

```
//
contract Timed{
  uint starttime = now;
  uint duration  = 30 days;
  modifier inTime {
    require(now<=starttime +
```

duration);
_;
}
(…Continua)

Da notare la speciale istruzione _; che è un segnaposto di quello che sarà la logica della funzione.

A questo punto dobbiamo utilizzare il modifier sulla fallback function in questo modo

function ( ) payable inTime{ }

Qui occorre spiegare due semplicissimi concetti. La funzione qui sopra si chiama “di fallback” e come il

lettore più attento avrà notato tale funzione non ha un nome. Significa che è il codice che viene eseguito di default quando il contratto riceve una semplice transazione che non indica il nome di una funzione. In altre parole se spedisco un semplice pagamento al contratto il contratto eseguirà il codice della funzione di fallback.

Seconda considerazione, si tratta ancora di una funzione che non fa nulla se non accettare una transazione senza nome e il corrispettivo valore in ether che viene spedito. Questa volta però oltre il modificatore payable abbiamo aggiunto anche quello inTime che garantisce che i pagamenti siano accettati solo entro la durata prevista dei

trenta giorni.

The King of Ether

Finora abbiamo visto come un contratto possa diventare un account con dei comportamenti speciali. In realtà i contratti possono avere delle funzioni con nomi e parametri. Vediamo un contratto che ha una funzione da noi denominata claim.

```
    contract King{
        address public king=address(0);
        uint public price=0.001 ether;
        string public name="";
        function claim(string _name)
payable {…code here…}
    }
```

Il codice del contratto non è riportato interamente ma potete vederlo in

Appendice. La logica del contratto è più o meno questa: chi invoca la funzione claim( ) e spedisce anche dell'ether al prezzo corrente price, può aggiudicarsi il diritto di vedere il suo nome come king. In tal caso il precedente king riceve la sua somma indietro ed il prezzo price è incrementato del 10%.

Le variabili king, price e name sono dichiarate public, questo significa che è possibile ispezionarle e vederne il valore dall'esterno, vedremo come.

Questo contratto è liberamente ispirato a the king of ether, un famoso smart contract descritto qui:

https://www.kingoftheether.com/

la ABI

Nella sezione precedente abbiamo visto come un contratto possa ricevere dei pagamenti speciali, o meglio transazioni che includono dei comandi invocando delle funzioni del contratto stesso. Supponiamo che il contratto King sia ormai in funzione nella blockchain. Come può un utente spedire un pagamento al contratto invocando la funzione claim( ) con il suo nome.

A questo scopo quando viene compilato un contratto viene generata anche un'interfaccia detta ABI che serve proprio per interagire con i contratti. Suona un po' complicato e di fatti lo è.

La ABI del nostro contratto è una definizione formale di comunicazione scritta in linguaggio JSON (uno standard molto usato per le comunicazioni tra applicazioni). Esso appare così:


```
    [{"constant":true,
      "inputs":[],
      "name":"name",
      "outputs"
[{"name":"","type":"string"}],
      "payable":false,
      "stateMutability":"view",
      "type":"function"},
     {"constant":true,
      "inputs":[],
      "name":"price", …
```

(Continuerebbe …) L'ho troncata dopo poche righe perché sarebbe troppo noiosa da leggere e commentare nel libro, se volete ne trovate una copia in Appendice. Quello che ci interessa è che grazie alla ABI possiamo istruire un wallet come MyEtherWallet ad interagire con un contratto che gira in blockchain. Basta inserire l'indirizzo del contratto, la sua ABI e sarà possibile invocare la funzione claim () e spedire qualche ether.

MyEtherWallet

New Wallet | Send Ether & Tokens | Swap | Send Offline | Contracts | Check TX Stat

Interact with Contract

Contract Address

0x55569a60236275d710c41ce534c5e56dede814fb

ABI / JSON Interface

```
[{"constant":true,"inputs":[],"name":"name","outputs":
[{"name":"","type":"string"}],"payable":false,"stateMutabi
{"constant":true,"inputs":[],"name":"price","outputs":
[{"name":"","type":"uint256"}],"payable":false,"stateMutab
{"constant":true,"inputs":[],"name":"king","outputs":
[{"name":"","type":"address"}],"payable":false,"stateMutab
```

Access

Nella figura precedente è mostrato come inserire la ABI e il contract address dentro MyEtherWallet. Nella figura successiva MyEtherWallet interpreta la ABI e ci mostra la funzione claim( ) e ci chiede di inserire l'argomento _name.

Read / Write Contract

0x55569a60236275d710c41ce534c5e56dede814fb

claim ▾

_name string

Davide

Strumenti per lo sviluppo

Prima di parlare degli strumenti per lo sviluppo e lancio dei contratti sulla blockchain dobbiamo rispondere a questa semplice domanda: **come facciamo a ottenere degli ether per fare delle prove e dei test?** Gli ether sono infatti i primi strumenti necessari per compiere ogni operazione sulla blockchain. Sono il cosiddetto cryptofuel. Certo usare gli ether "veri" potrebbe essere uno spreco quando si fanno delle prove e dei test, soprattutto se siamo imparando. E' meglio ottenere degli ether "falsi", ovvero lavorare su una chain di test. Per gli esempi qui illustrati fra le varie chain disponibili

scelgo la testnet Kovan. Il motivo della scelta è molto semplice, è stata la testnet per la quale ho potuto ottenere con maggior facilità degli ether.

Gli ether della chain Kovan non hanno valore sul mercato, quindi possiamo spenderli con una certa leggerezza, non sprecarli comunque. Infatti anche gli ether "finti" hanno una cerca scarsità e bisogna stare attenti a non finirli. Per ottenerli ci sono faucet o forum. Un esempio di faucet è https://gitter.im/kovan-testnet/faucet ma naturalmente è meglio cercare con Google dato che questi servizi nascono e muoiono continuamente.

Una volta ottenuti gli ether possiamo concentrarci sul nostro contratto. Prima

di tutto dobbiamo usare un client per la rete Ethereum che ci permetta di creare transazioni e contratti e firmarli con le chiavi private che andremo a creare. A questo scopo possiamo installare un wallet locale come Mist, Parity o un'estensione per il browser come Metamask.

Riferiamoci per ora al client Parity e per quanto riguarda le istruzioni di installazione queste sono reperibili dal sito ufficiale parity.io

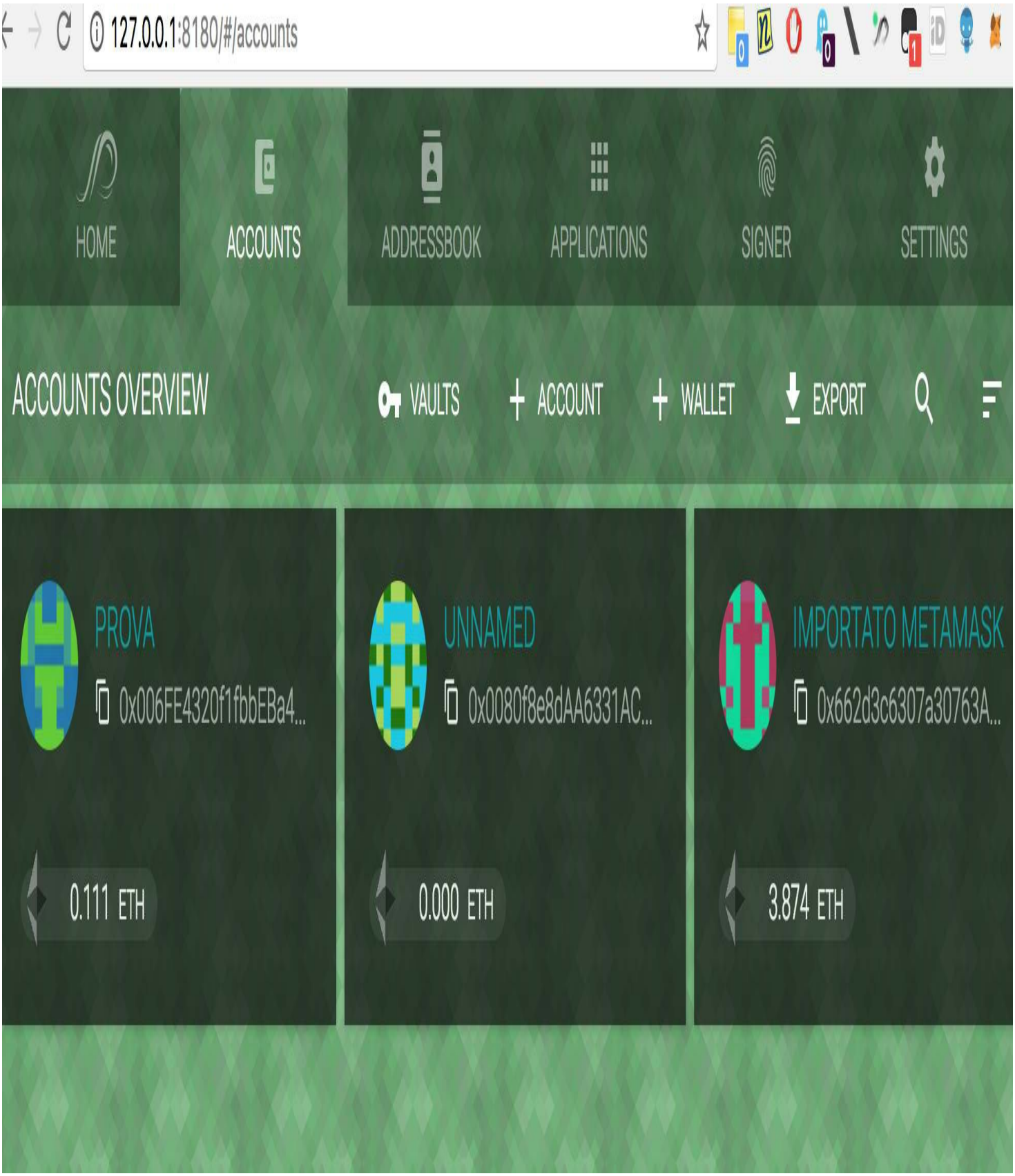
Una volta installato parity dovrà essere lanciato in modo che sincronizzi la blockchain locale, se è la prima volta che viene usato ci metterà un bel po' di tempo. Diciamo un paio d'ore ma

dipende dalla velocità di connessione. Lanciare parity sulla testnet Kovan è molto semplice

parity --chain kovan

A questo punto bisogna aspettare …

Una volta che la blockchain è in sync possiamo occuparci del resto. Prima di tutto creiamo qualche account usando parity. In generale parity dispone di un'interfaccia utente visualizzabile nel browser collegandosi all'indirizzo della macchina locale 127.0.0.1:8180


127.0.0.1:8180/#/accounts
HOME
ACCOUNTS
ADDRESSBOOK
APPLICATIONS
SIGNER
SETTINGS
ACCOUNTS OVERVIEW
VAULTS
ACCOUNT
WALLET
EXPORT
PROVA
0x006FE4320f1fbbEBa4...
0.111 ETH
UNNAMED
0x0080f8e8dAA6331AC...
0.000 ETH
IMPORTATO METAMASK
0x662d3c6307a30763A...
3.874 ETH

Creare nuovi account è molto semplice, basta premere il tasto +ACCOUNT e seguire le istruzioni. E' possibile anche importare un account da un altro wallet come ad esempio Metamask. Gli importi di ogni account sono in ether ovviamente e su rete Kovan in questo caso.

Parity consente anche di sviluppare dei nuovi contratti in solidity. Sebbene non rappresenti una soluzione completa per lo sviluppo di applicazioni decentralizzate è comunque molto pratico e semplice da utilizzare.

In alternativa a Parity ci sono soluzioni come Remix, un IDE

completamente online che però consente di gestire i propri file locali e poi due soluzioni sicuramente più complete e professionali come Truffle e Populus che consentono di gestire il ciclo di vita dei contratti in modo professionale, dall'editing alla creazione di testcase. Truffle in particolare sta diventando un vero e proprio framework standard e sempre più progetti solidity lo stanno adottando.

I Token e lo standard ERC20

I token prima di tutto sono degli smart contract che in generale possono essere implementati con una logica che preveda che un possessore di token possa distribuirli ad altri indirizzi e mantenere un tabella di tutti saldi di ogni possessore. Tuttavia, dato che è molto facile scrivere del codice non sicuro, sono state proposte delle convenzioni o meglio dei pattern per realizzare il token perfetto o perlomeno non troppo vulnerabile. Partiamo dall'interfaccia ERC20. Un'interfaccia è un insieme di funzioni che ci aspettiamo un contratto di tipo ERC20 debba implementare. La sintassi e il comportamento delle

funzioni ERC20 è descritta qui sotto.

function totalSupply()

restituisce l'ammontare totale dei token in circolazione.

function balanceOf(account)

restituisce il saldo dell'account con indirizzo address

function transfer(beneficiario, ammontare)

Trasferisce un certo ammontare di token da chi invoca la funzione ad un account beneficiario. Se chi invoca la funzione non dispone di token sufficienti la transazione fallisce e niente viene

modificato nello stato.

L'interfaccia ERC20 in realtà definisce anche altre funzioni che non analizziamo qui per brevità. Una copia completa dell'interfaccia ERC20 è riportata in Appendice per comodità del lettore.

In pratica con il nostro client Parity possiamo subito creare un token ERC20. Infatti Parity fornisce già degli snippet di codice pronti per creare un token con pochi clic di mouse. Tra questi esempi c'è anche HumanStandardToken.sol che possiamo compilare e "lanciare" nella blockchain.

In questo caso ci verrà chiesto di configurare, cioè passare quei parametri del costruttore visibili nel codice

CONTRACT PARAMETERS

contract details — 2 contract parameters

Choose the contract parameters

_initialAmount: uint256

100

_tokenName: string

toki

_decimalUnits: uint8

0

_tokenSymbol: string

toki

Nel nostro caso abbiamo creato un contratto di nome Toki, come il leggendario guerriero manga, con un supply di 100 TPS e nessuna cifra decimale consentita.

Una volta in esecuzione possiamo andare a ripescarlo da qualsiasi blockchain explorer.

Ad esempio, etherscan lo vede così

KOVAN

Etherscan

The Ethereum Block Explorer

ERC20-TOKEN toki

TokenTracker Summary

Total Supply:	100 toki
Value per Token:	$0.00
Token Holders:	2 addresses
No.Of.Transfers:	1

EtherScan mi segnala che ci sono due detentori del mio token Toki.

Token e tokensale

Abbiamo creato il nostro token contract, come facciamo ora a distribuire i token?

A quanto pare non esiste una funzione nel nostro contract ERC20 che permette di emettere token in cambio di ether. Ci saremmo aspettati una funzione di fallback tipo payable e invece non è prevista nello standard.

Non è prevista, ma questo non significa che noi non possiamo aggiungerla. In generale possiamo fare quello che ci pare con il nostro contratto di token. Una possibilità è quella di

assegnare tutti i token inizialmente al creatore del contratto. Questo account, manovrato da un utente, può decidere cosa fare dei token e a chi assegnarli o venderli.

Tuttavia nel caso di una token sale ad esempio per una ICO non ci aspettiamo che manualmente qualcuno raccolga gli ordini e poi spedisca i token. Siamo nel mondo degli smart contract quindi dobbiamo automatizzare questo processo e non vogliamo cassieri umani che leggono le transazioni di ether in ingresso e spediscono i token agli acquirenti.

Il nostro contratto token potrebbe avere una funzione di fallback payable e gestire semplicemente così

```
function() payable {
        uint qty = msg.value / pricePerToken;
        bool success = transfer(msg.sender,qty);
        if (!success) throw;//out of token
    }
```

Ricordiamoci che la funzione di fallback è quella funzione che viene attivata quando il nostro contratto riceve un semplice pagamento che non indica il nome di una funzione specifica da

eseguire.

msg.value invece è la quantità di ether (quindi denaro) che la transazione sta trasportando con sé. In pratica la quantità di coin trasmessi da chi invoca questa funzione.

Il guaio con la soluzione di mettere la token sale ed il token ERC20 nello stesso contract è che una volta in esecuzione non possiamo cambiare. Quindi se poi decidiamo che i prezzi dei token non ci vanno bene o vogliamo implementare una politica dinamica di prezzo allora dovremo rinunciarci.

Un modo più flessibile è quello di avere due contratti, uno il contratto di

token vero e proprio ed un altro contratto che fa la token sale, ovvero che implementa una funzione payable di fallback. Questo secondo contratto di token sale potrebbe essere rimpiazzato con uno nuovo se vogliamo inserire delle migliorie, senza però cancellare o invalidare il contratto di token ERC20.

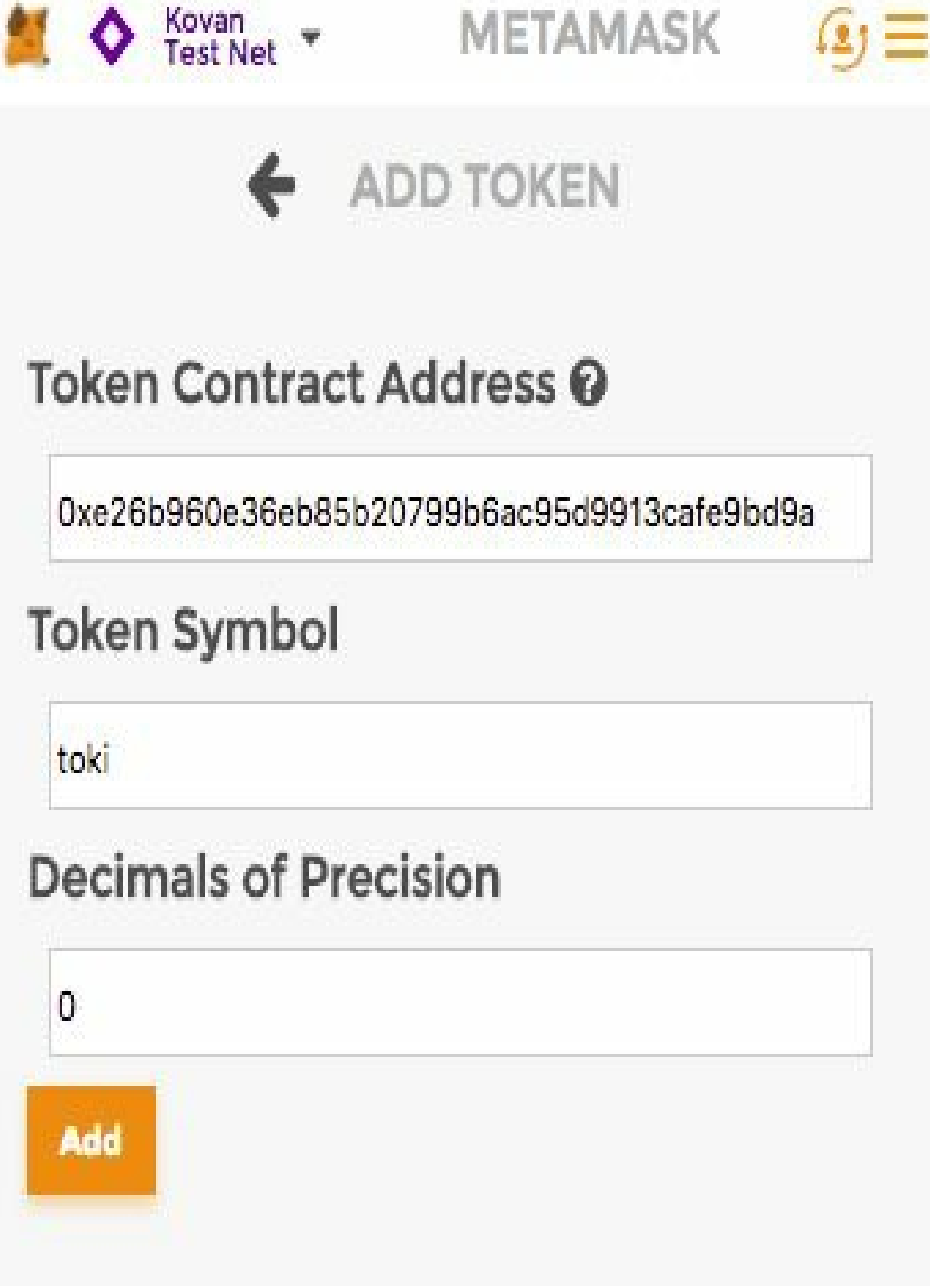
Una pratica abbastanza comune è quella di riutilizzare il codice del progetto Open Zeppelin (lo trovate su Github) che fornisce contratti base e utility per la gestione di wallet, token e token sale.

Strumenti per gestire i token

Gli strumenti per interagire con i token ERC20 sono essenzialmente i wallet. Metamask consente di inserire la descrizione di un token contract di cui si dispone dell'address e della sigla in modo molto semplice. Sfortunatamente Metamask non consente di spedire i token per il momento, ma solo di verificarne il saldo.

Se vogliamo gestire i nostri token con un wallet super semplice come Metamask non dobbiamo far altro che cliccare su "Add Token" e inserire il token contract address come in figura. Potremmo vedere il balance di tutti i

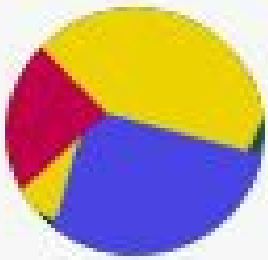
nostri token direttamente, nel nostro esempio vediamo **ben** 11 toki nel nostro wallet.


Kovan
Test Net
METAMASK
ADD TOKEN
Token Contract Address
0xe26b960e36eb85b20799b6ac95d9913cafe9bd9a
Token Symbol
toki
Decimals of Precision
0
Add



Kovan
Test Net

METAMASK



Account 1

0x8387A...

0.811970 ETH

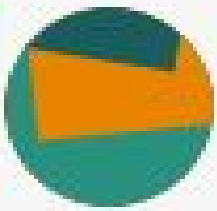
BUY SEND

SENT TOKENS

You own 3 tokens

ADD TOKEN

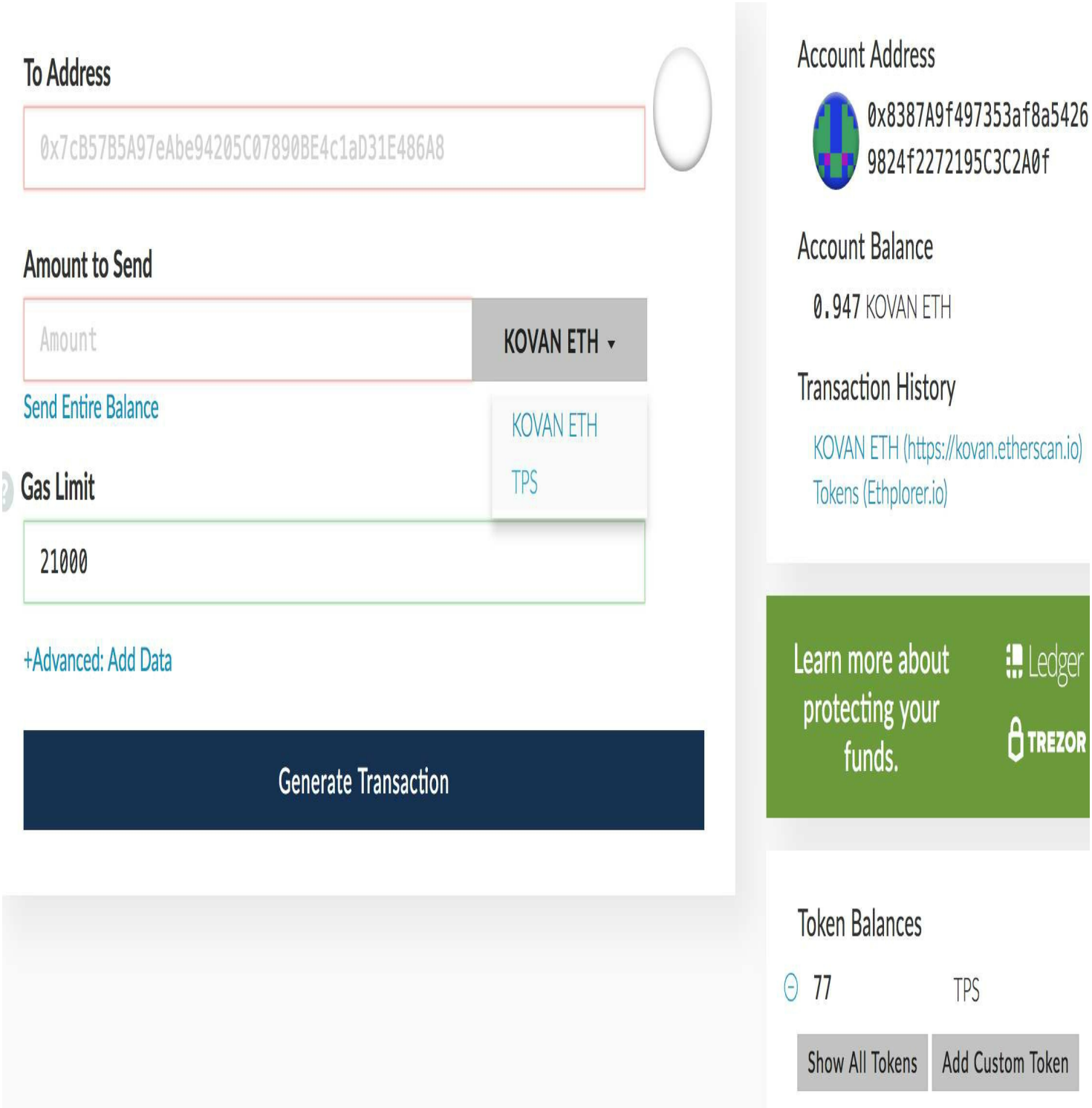
77 TPS



9999 DDT

11 toki

MyEtherWallet è invece un wallet più completo per interagire con i token. Si possono osservare i saldi e spedire facilmente. Nell'esempio sotto il token con sigla TPS viene gestito tramite MyEtherWallet.



Naturalmente anche i client Parity e Mist essendo i più completi possono gestire facilmente i token. Si rimanda

per questo alla loro documentazione.

Un altro strumento utilissimo per l'utilizzatore esperto di Ethereum e dei suoi token è senz'altro Etherscan. Attraverso questo sito web si possono esplorare indirizzi, transazioni, blocchi, contratti e tutti gli elementi contenuti nella chain. Una sezione apposita del motore di ricerca consente di cercare i token ERC20 presenti e vederne tutti i possessori e ogni transazione. Nella figura sotto viene cercato un token di nome tokenpolis.

KOVAN Etherscan
The Ethereum Block Explorer

ERC20-TOKEN tokenpolis (*Unverified)

TokenTracker Summary

Total Supply:	1,000 tokenpolis
Token Holders:	5 addresses
No.Of.Transfers:	12

Token Transactions | Token Holders | Read Smart Contract

TokenHolders Chart

A total of 5 Token Holders found

Rank	Address
1	0x16a98dc36f97005487dece406b2875a2d02b5eec
2	0x8387a9f497353af8a54269824f2272195c3c2a0f
3	0xf1a2ad204d5a175bf4ac4539c8e9fb7d448e5a6b
4	0x006fe4320f1fbbeba4a6df4c72173186962bb71b
5	0x6c22f997e8aae4567c949c444b22b4079b30c37f

Considerazioni finali

Abbiamo a volte citato il paradossale pensiero che dice che gli smart contract non sono né smart né contract.
Naturalmente un giurista vede i contract Ethereum troppo distanti dal concetto di contratto definito dal codice civile. Per un programmatore, invece sono semplici

programmi, chiamarli contract o class forse non farebbe nessuna differenza. Infine per un cripto-anarchico invece l'idea di una macchina incensurabile e inarrestabile che agisce in modo equo e condiviso dalle parti sembra un'incarnazione perfetta del concetto di smart contract.

Personalmente la parola *smart* la trovo un po' abusata, con smart ci dovremmo riferire a qualcosa di intelligente, ma un po' per estensione, tutto quello che è innovativo finisce per diventare smart. Di fatto sarebbe più corretto dire contratto crittografico o contratto digitale.

Digitale non significa digitalizzato, non è che prendiamo un contratto cartaceo e lo digitalizziamo questo diventa uno smart contract. Digitale significa che può fare uso delle tecniche digitali (leggi numeriche) per consentire alcune proprietà: ad esempio l'autenticità. Questo si ottiene attraverso la crittografia.

Smart in realtà dovrebbe significare "capace di ragionare con dati incompleti" mimando in qualche modo il ragionamento o l'intuizione umana. Qualcosa più legato all'intelligenza artificiale che alla certezza crittografica. Ma questo è esattamente quello che non vogliamo nei contratti digitali: la

discrezionalità dell'interpretazione che porta facilmente alla corruzione e all'ingiustizia. Ma qui ancora una volta siamo nel campo delle opinioni sul significato del termine *smart*.

Per curiosità, cos'è un contratto secondo il codice civile italiano?

Il contratto è l'accordo di due o più parti per costituire, regolare o estinguere tra loro un rapporto giuridico patrimoniale.

Insomma, serve definire le parti, che nell'ordinamento italiano saranno persone fisiche o giuridiche mentre nel

caso degli smart contract ci sono solo account e le loro chiavi private. Queste potrebbero essere possedute da un minore, da un robot, da uno scimpanzé e lo smart contract continuerebbe comunque a funzionare perfettamente. Il contratto stesso su Ethereum può possedere una dotazione finanziaria. Insomma un contratto su Ethereum è più che altro un agente artificiale (non intelligente però) dotato di fondi propri che può interagire con altri attori anch'essi dotati di chiavi crittografiche ed account sulla blockchain. Inoltre sempre secondo il codice civile

L'oggetto del contratto deve essere possibile, lecito, determinato o

determinabile

Ammesso di poter identificare un oggetto nel caso degli smart contract, i concetti di possibile, lecito, determinato devono essere traslati nel vocabolario della blockchain.

In blockchain non esiste il concetto di possibile/impossibile ma quello di valido o non valido. La blockchain è fatta di transazioni, esse sono possibili se la matematica lo consente (esempio se hai 1 ether non puoi spenderne 1.5). Le transazioni sono verificate e successivamente inserite nel registro globale.

Lecito significa a norma di legge, ma in questo caso quale legge? Per la blockchain esiste il protocollo formale e non conosce la giurisdizione in cui opera il nodo della rete.

Code is law

Anche la definizione qua sopra però mi causa qualche grattacapo. Infatti non esiste software senza bug, quindi un bug come lo consideriamo? Parte della legge o un motivo per riscrivere la legge?

La natura degli smart contract sembra più quella di agenti automatici che di contratti intesi come accordi. Tali agenti obbediscono ad un codice sorgente che

ne stabilisce la logica e le condizioni.

Quindi, se non sono contratti a cosa servono e cosa si può fare? Ci sono diverse applicazioni:

- Aziende decentralizzate, attraverso un fondo comune soggetto a regole con cui i partecipanti possono votare per le spese da sostenere e regole per i dividendi
- Testamento, stabilisce che una somma venga spedita al verificarsi della morte di una persona.
- Regalo di compleanno, in cui una somma viene spedita solo al compimento dei 18 anni.

- Assicurazioni, un contratto di assicurazione che risarcisce automaticamente al verificarsi di un sinistro.
- Diventare king of ether

e molti altri ….

Quindi qual è la peculiarità degli smart contract rispetto ai normali programmi del calcolatore? Beh, possiamo dire che la loro forza sta nel fatto che nessuno può spegnerli: una volta avviati, se la terminazione non è prevista e codificata nel loro codice sorgente, allora saranno per sempre in esecuzione. Si potrebbe suggerire che per spegnere un programma basta

staccare la spina al computer che lo esegue, ma qui sta il bello ed il brutto della blockchain: il programma viene eseguito contemporaneamente ed in modo identico da tutti i nodi della blockchain che sono migliaia. Il bello? Nessuno può staccare la spina. Il brutto? Nessuno può staccare la spina, inoltre è l'esecuzione più energeticamente inefficiente che possiamo immaginare. Ma ricordiamo che la blockchain è progettata per la sicurezza in un ambiente estremamente ostile dove tutti i partecipanti sono hacker esperti, malintenzionati e anonimi che provano ad imbrogliare mentre indossano la maschera di Guy Fawkes, e nonostante questo la sicurezza e l'onestà delle

transazioni è garantita. La blockchain non è pensata per essere energeticamente efficiente.

Se un avversario ostile staccasse la spina al 50% dei nodi (nodi = computer) in poche ore il protocollo si auto-adatterebbe rendendo appetibile a molti nuovi potenziali nodi il loro ingresso. Di fatto i nodi sono sempre in competizione fra loro e se la competizione è più facile questo fa si che immediatamente altri nodi si aggiungano.

Smart contract: take-aways

Ethereum come Bitcoin usa un meccanismo di consenso della prova di lavoro che consuma molta energia, ma presto migrerà ad un diverso protocollo detto proof of stake.

Diversamente da Bitcoin, Ethereum prevede che si possano scrivere dei

contratti complessi sulla blockchain. Se Bitcoin è l'internet del denaro, Ethereum è il denaro programmabile.

Si chiamano smart contract, ma sono dei veri e propri programmi. Saranno i programmatori a scriverli e non gli avvocati.

I contratti sono fatti di istruzioni e ogni istruzione ha un costo detto GAS, il GAS va pagato ai miner e si paga in ether.

I wallet di Ethereum oltre che spedire pagamenti possono interagire con i contratti. Il protocollo per "parlare" con i contratti si chiama ABI.

Tra i contratti ne esiste un tipo speciale detto token. Ognuno di noi può così definire la sua currency e farla girare nella blockchain di Ethereum.

Le ICO

Introduzione

What if issuing a token is the new "listening on port 80" ?

ICO significa Initial Coin Offering e significa emettere token ancora senza valore in cambio di ether o bitcoin. Ma allora perché molti

progetti hanno raccolto anche centinaia di milioni di dollari con questo meccanismo?

Ormai le ICO sono diventate un vero e proprio fenomeno all'interno del mondo hi tech. Siamo probabilmente vicini allo scoppio di una vera e propria bolla.

Ma bolla a parte quale lezione possiamo estrarre da queste operazioni che avvengono all'insaputa dei mass media e che stanno comunque plasmando una parte del futuro di tutti.

Da quando Ethereum ha cominciato a prevalere come piattaforma per gli smart

contract si sta verificando che la sua killer application è proprio l'emissione di nuovi token. Chiamiamoli pure domain specific tokens.

Torniamo alla domanda iniziale. Ma alla fine a cosa possono servire i token? In primis servono sicuramente a chi li emette come sorgente di finanziamento, un **mero crowdfunding**. I founder cedono token in cambio di cripto più solida come bitcoin o etere. Il capitale viene raccolto per sviluppare un servizio o un prodotto. Ma poi chi si trova i token in mano cosa se ne fa? Nei casi più importanti un po' di speculazione, infatti non è raro che i token emessi in crowdsale vengano poi

listati in mercati telematici dove possono essere scambiati con altre criptomonete e con altri token. Ricordiamo che la prima ICO di qualche rilievo è stata proprio quella di Ethereum: 2000 ether per 1 BTC. La speculazione in tal caso è andata benissimo dato che 1 BTC nel 2014 valeva circa 500 dollari e 2000 ether oggi varrebbero circa 500,000 dollari. Un fattore mille in tre anni. Non male

Ma la speculazione non è tutto, o almeno non dovrebbe esserlo. I token dovrebbero anche attestare il diritto di accesso e utilizzo di un determinato servizio. Nel caso di Ethereum, la sua

currency detta ether serve per pagare le fee alla rete di miners che esegue i contratti e valida le transazioni inserite dagli utenti.

Una caratteristica delle ICO è la velocità. L'impressionante velocità di raccolta dei capitali

Vediamo alcuni casi interessanti.

Basic Attention Token (crowdsale $35M in 30 secondi)

I BAT sono i Basic Attention Token e dovrebbero servire a ridefinire un mercato dell'advertisment su Internet per rompere il duopolio Google

Facebook. Il principio di funzionamento è basato su un nuovo browser chiamato Brave che consente in modo privato e non tracciato di remunerare direttamente i contenuti nel caso di contenuti premium, oppure di remunerare attraverso i click sulla pubblicità ma senza passare da un ente centrale come Google o Facebook che controlla e gestisce questi flussi. Il tutto funzionerebbe in modo decentralizzato e i BAT sarebbero usati per la monetizzazione istantanea.

IOTA (capitalizzazione $1B)

Un secondo caso interessante è quello di IOTA che fornisce un sistema

di nanopagamenti senza fee e senza utilizzare una dispendiosa blockchain proof-of-work. Tale sistema avrebbe come caso d'uso l'IoT (Internet of things). Avere questi token dovrebbe servire come "benzina" per la realizzazione di applicazioni che integrano ogni tipo di device. Infatti uno dei problemi nell'uso delle blockchain "a la Bitcoin" per i micropagamenti è il costo energetico assolutamente proibitivo per la rete. Una transazione Bitcoin richiede molta energia e non vale la pena usarlo per piccoli pagamenti. Tuttavia con le evoluzioni di microchannel payments, segwit e lightning networks anche Bitcoin potrebbe superare questo ostacolo.

Bancor (crowdsale da $150M)

Il nome Bancor viene da una moneta ideale pensata nientemeno da Keynes come una unità standard sulla quale misurare tutte le altre monete. Un po' quello che fa il dollaro, ruolo che il dollaro si è conquistato a colpi di guerre e dominazioni e quindi il cui supply è nelle mani di un governo. Il bancor era un modello di moneta "condivisa" e internazionale che però non ha mai preso piede. Bancor oggi è invece una tecnologia decentralizzata basata su Ethereum che ha emesso il suo token con sigla BNT. L'idea della Bancor network è quella di facilitare l'emissione di

domain specific token e di rappresentare lo standard per tutti questi nuovi asset garantendone la liquidità indipendentemente dalla dinamica di domanda/offerta. Ma perché dovremmo volere che piccoli domain specific token debbano godere di un prezzo stabile e un supply decente?

Arriveremo al punto che ogni parchimetro, ogni negozio, ogni società di persone vorrà emettere il suo token?

E troppo presto per dare una risposta. Intanto i token vengono creati a centinaia. Alcuni progetti hanno senso, altri sono interessanti ma poco

sostenibili e alcuni sono vere e proprie truffe.

Quelli che viviamo sono sicuramente tempi molto interessanti e bisogna approfondire questi argomenti. **Che tu sia un investitore, un curioso, uno che vuole lanciare la sua ICO o un bitcoiner che vuole investire in qualche ICO altrui ti invito a leggere il seguito.**

Il white paper

Quando si vuole lanciare una ICO, un passo essenziale è quello di pubblicare un white paper che descriva gli obiettivi del progetto e a cosa serve il token. In genere una ICO serve per finanziare lo sviluppo di una piattaforma tecnologica in cui il token è un elemento essenziale al funzionamento. Tuttavia non è sempre

così, non esistono regole certe su come organizzare e portare avanti una ICO di successo.

Stesso discorso vale per il white paper? Esiste una ricetta per scrivere un white paper vincente? Direi di no. Per fortuna nessuno ha ancora dettato lo schema standard e questo rimane ancora un settore dove ognuno usa lo stile che gli pare. Nessuno auspica **il formato europeo del white paper.**

Il white paper è un genere "letterario" che nasce in ambienti accademici e scientifici. Una specie di articolo che non punta alla pubblicazione su rivista ma piuttosto

alla massima diffusione attraverso la rete. Esiste nel mondo del business un tipo di documento che si può considerare parallelo a quello del white paper tecnico: il business plan.

In effetti il business plan ha un focus sugli aspetti più commerciali e sulla capacità remunerativa mentre il white paper in genere si concentra su aspetti tecnici più che su quelli economici e finanziari.

Tuttavia come la ICO rimpiazza la IPO nell'ambizione di raccolta di capitali, il white paper deve essere anche un buon business plan poiché **chi compra i token vuole soprattutto una**

cosa: finanziare un bel progetto che abbia un valore ma anche speculare sul prezzo del token acquistato.
Questa aspettativa, che sia essa celata o manifesta, influenza moltissimo anche la possibilità che gli organismi di vigilanza si mettano a ispezionare la vostra ICO.

Va da sé che tanto più un progetto si mostra economicamente sostenibile tanto più i token andranno a ruba e chi ha investito all'inizio potrebbe trovarsi con un bel gruzzolo in mano.

In sostanza, riguardo il white paper, bisogna considerare che la platea dei lettori potrebbe non essere tecnica né esperta in questioni finanziarie. Questa

differenza è rilevante rispetto ai business plan che in genere sono rivolti a investitori istituzionali. Quindi la regola d'oro è **cercare di farsi capire da tutti** se si hanno delle buone idee.

Nel white paper bisogna chiarire essenzialmente questi punti:

- A cosa serve il token? Garantisce un diritto all'accesso di qualche funzionalità del progetto? Oppure rappresenta un credito od una qualche forma di equity? Oppure un mix delle due cose o magari nessuna delle due?
- Ammesso che il token abbia una funzione tecnica, qual è la sua

scarsità, ovvero in cosa si distingue la generazione del token e la sua scarsità rispetto alla scarsità della moneta principale come eth o btc? Bisogna considerare che se il token non ha una scarsità specifica potrebbe non avere senso la sua esistenza ed essere immediatamente rimpiazzabile dall'ether o dal bitcoin.

- Posto che il white paper chiarisca la funzione del token, sia questa tecnica o puramente finanziaria, è importante che il progetto fornisca una soluzione praticabile ad un problema reale. Praticabile significa anche sostenibile nel tempo da un punto di vista

economico e quindi che faccia breccia in un mercato che fornisca le risorse necessarie. Questo dovrebbe essere il requisito principale di un white paper ovvero convincerci del valore del progetto e del possibile impatto nella realtà.

- Ammesso che il progetto stia affrontando un problema rilevante in modo appropriato bisogna chiarire quali risorse si mettono in campo sia in termini di risorse umane e competenze che eventualmente attrezzature o impianti necessari. Banalmente, chi sono le persone che lo svilupperanno? Ci possiamo fidare

delle loro capacità?

- Infine, posto che tutti i punti precedenti siano chiariti, bisognerebbe spiegare come il progetto possa subire un audit esterno per garantire che lo svolgimento delle attività sia quello promesso. In parole povere - **Cosa state combinando con i nostri soldi?**

Presale

In generale prima della ICO si attraversa un periodo detto di presale o pre-ICO che risulta essere un periodo frenetico in cui si devono ottenere sostanzialmente due obiettivi: arrivare con un'implementazione efficiente e solida della raccolta di capitali, il che significa aver predisposto gli opportuni wallet,

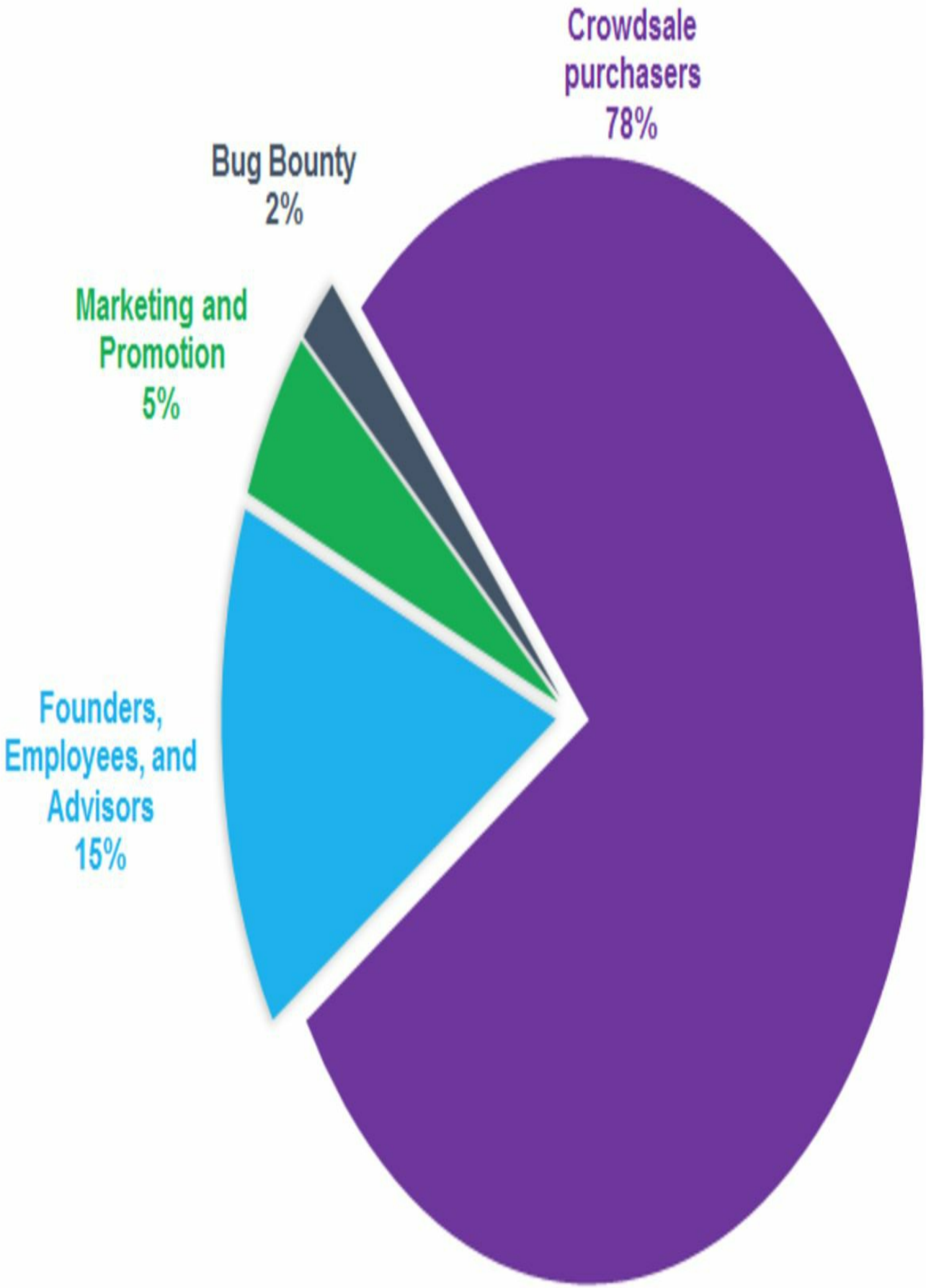
smart contracts e altri schemi necessari; e sviluppare le attività di marketing e dissemination in modo che la nostra ICO sia pubblicizzata il più possibile raggiungendo i potenziali investitori in ogni angolo del web.

La pre-sale potrebbe anche svilupparsi attraverso un più tradizionale promozione presso gli investitori istituzionali con lo scopo di raccogliere i capitali necessari per il lancio della successiva ICO. Il marketing costa e fare una ICO non fa eccezione. Non siamo tutti dei Vitalik e più ICO vengono lanciate nel mondo più difficile sarà far emergere il proprio progetto agli occhi dei potenziali

investitori.

In genere di tutti i token previsti per l'emissione una percentuale più o meno consistente viene distribuita nella fase di pre-sale secondo delle modalità del tutto peculiari che cambiano da ICO a ICO. Per esempio possono essere distribuiti token per il team e per gli investitori tradizionali, possono essere distribuiti in forma di bounty per tutti quelli che fanno follow, like e retweet sul canale Twitter del progetto o fanno "like" o "share" sui post della pagina Facebook ufficiale. Altre possibilità sono ad esempio delle bounty per la traduzione del materiale pubblicitario in altre lingue. Nella figura che segue è riportata

la ripartizione possibile dei token in funzione delle varie attività.


Crowdsale purchasers 78%
Bug Bounty 2%
Marketing and Promotion 5%
Founders, Employees, and Advisors 15%

Meccanica della crowd sale

Proviamo a esplorare alcune delle ICO più interessanti e qual è stata la dinamica con cui si sono svolte.

Introduzione

Ogni ICO stabilisce le regole con le quali raccoglie i capitali. Non esiste un canone perché siamo in un territorio nuovo, inesplorato e ricco di rischi e opportunità. Tra le scelte del team ci sono anche quelle sul funzionamento pratico del finanziamento: quando parte, quando finisce, quanti token sono distribuiti e quanti sono tenuti dal team, a quanto ammonta il minimo ed il massimo della raccolta e quali sono le regole per interrompere la vendita dei token in anticipo se necessario.

Vitalik Buterin nei suoi scritti

suggerisce due requisiti nel design di una ICO. Il primo riguarda la certezza della valutazione del pacchetto di token acquistato da un ipotetico investitore. In altre parole se noi acquirenti acquistiamo l'1% dei token ci aspettiamo che questo pacchetto non venga diluito dall'emissione di successivi token.

Un secondo requisito è la massimizzazione della partecipazione, ovvero una ICO ben progettata dovrebbe permettere a tutti quelli che lo desiderano di poter acquistare i token. Purtroppo i due requisiti presentati sono in tensione come è facilmente osservabile. Se infatti fosse garantito a tutti l'acquisto di token in teoria

saremmo costretti ad incrementare il supply dei token stessi andando a diluire i pacchetti già acquistati dai precedenti partecipanti.

Altre scelte di design riguardano il tetto all'ammontare del capitale raccolto e quello di mantenere o meno a propria disposizione una grande quantità di token, impersonando di fatto il ruolo di banca centrale.

La meccanica di queste token sales spesso segue delle strategie mutuate dal mondo pre-blockchain e sono magari varianti delle Dutch Auction e delle Vickrey Auction.

Nelle sezioni successive analizziamo alcuni casi concreti come Maidsafe, Ethereum, BAT, Bancor e Gnosis.

Maidsafe

Maidsafe è una piattaforma decentralizzata che ha raccolto $7m in cinque ore. Era possibile acquistare i token attraverso pagamenti in bitcoin o mastercoin (un'altra valuta associata al progetto Omni) con un leggero vantaggio a favore della seconda opzione. Questo ha causato un picco di richiesta del mastercoin che ha praticamente raddoppiato il prezzo, prezzo poi crollato alla chiusura della crowdsale. La corsa ad acquisire i mastercoin in cambio dei loro bitcoin ha significato per diversi utenti incassare una perdita fino al trenta percento. La lezione principale di questa esperienza è che

non bisogna offrire una ICO con diverse valute. O meglio, il prezzo deve essere indicato con una sola valuta, ad esempio in bitcoin, e poi può essere aperta ad altre forme di pagamento ma il prezzo è univoco nella valuta scelta.

The Ethereum sale

Ethereum è stata probabilmente una delle prime importanti crowdsale della storia. Cronologicamente seconda solo alla ICO di Mastercoin. Gli ether venivano descritti come il carburante per l'esecuzione degli smart contract.

La vendita degli ether è stata di tipo uncapped, in pratica non c'era un limite alla quantità di token ma il limite era nella durata temporale di accesso all'acquisto. Tale finestra di acquisto è stata di 42 giorni e nei primi 14 giorni il prezzo era fissato al minimo di 2000 ether per 1 bitcoin. Nei successivi 28 giorni il prezzo ha subito un aumento lineare, questo per premiare gli early

adopters.

Il fatto che non sia stato imposto un limite al numero di token ha introdotto un'incertezza sul valore dei token stessi dato che è venuta a indebolirsi la caratteristica di scarsità.

La politica inflazionistica di Ethereum resta tuttora un mistero. I detrattori considerano questa troppo legata alla figura di Vitalik Buterin che può fare il bello e il cattivo tempo.

BAT sale

Diversamente da Ethereum, BAT (e altri token) hanno invece optato per la tipologia capped, ovvero la quantità totale di token è nota e limitata da subito. Questo crea un'impressione di scarsità ed un forte incentivo all'acquisto sin dalle prime fasi, il cosiddetto FOMO o Fear Of Missing Out.

Un progetto come BAT che ha suscitato grande interesse per i suoi contenuti tecnici è stato in grado di completare la vendita di tutti i token in soli 30 secondi (due blocchi nella

blockchain) raccogliendo una quantità di ether equivalente alla cifra di 35 milioni di dollari. Alcune statistiche riguardo questa vendita sono impressionanti. Durante i due blocchi sono stati spesi oltre 15 mila dollari di fee, solo 185 tentativi di acquisto sono andati a buon fine mentre oltre 10000 sono falliti. La capacità della piattaforma Ethereum è stata messa a dura prova per circa tre ore dall'inizio della vendita.

Le fee sono l'unico strumento competitivo per un acquirente che vuole battere gli altri sul tempo e ottenere i token prima della chiusura della vendita. E' prevedibile che in situazioni simili in futuro gruppi organizzati di miner

possano acquisire i token battendo sul tempo tutti rifiutando le transazione degli utenti non miner, e rivendendolo successivamente a prezzi molto più alti.

Gnosis

Per alleviare l'effetto di corsa al token delle vendite con cap è stato pensato il meccanismo di asta olandese inversa (reverse dutch auction). In questo tipo di vendita abbiamo un limite espresso in valuta fiat ed il numero di token emessi dipende dalla lunghezza della vendita.

In pratica il prezzo d'acquisto parte alto e poi ad ogni blocco della chain viene abbassato. Nel frattempo i potenziali acquirenti possono decidere se inviare o meno i loro ether, senza sapere però quale sarà la valutazione finale del token. Nel momento in cui si raggiunge il cap desiderato il prezzo del

token viene stabilito per tutti e sarà in funzione di quanto tempo l'asta è rimasta aperta. In generale se T è il numero di blocchi il prezzo finale sarà inversamente proporzionale e dunque si muoverà come 1/T.

La strategia di chi acquista dovrebbe essere semplice e di tipo win-win. Se come acquirente non accetto di spendere il prezzo corrente aspetterò fino a che il prezzo scenderà al di sotto della soglia che soggettivamente reputo onesta. I casi sono due: se l'asta si chiude prima perderò la possibilità di acquistare ma essendo il prezzo troppo alto secondo la mia opinione il non acquisto sarà un win. Se l'asta si chiude sotto il prezzo

da me stimato allora acquisterò ad un prezzo conveniente e l'acquisto sarà un win.

Nel caso di Gnosis questa strategia non ha però funzionato. Per la cosiddetta "fear of missing out" gli acquisti si sono concentrati nel primo giorno. Questo proverebbe l'irrazionalità degli acquirenti e la loro manipolabilità. Tuttavia il prezzo dei token Gnosis non solo ha tenuto nelle settimane successive la vendita ma ha perfino aumentato il suo valore. In poche parole gli acquirenti si sono dimostrati irrazionalmente corretti.

Un altro aspetto da tenere in

considerazione è che Gnosis non solo ha aggiustato il prezzo in base alla durata della vendita ma anche il supply (il totale di token distribuiti). In altre parole Gnosis ha tenuto per se una grande percentuale di token e ha bloccato questi per un anno con l'obiettivo di mantenerne stabile il prezzo.

Il fatto di mantenere nelle casse di Gnosis una grande percentuale dei suoi token mette Gnosis stessa nella posizione di poter manipolare i prezzi e agire come una sorta di banca centrale della sua valuta.

Bancor

Le caratteristiche della vendita di Bancor sono state descritte dal team in questi termini:

La vendita dura al massimo 14 giorni
Esiste un cap segreto che verrà rivelato solo quando sarà raggiunto l'80% del cap stesso.

Durante la prima ora saranno accettati tutti i fondi versati, anche in eccesso rispetto al cap. In tal caso la vendita terminerà esattamente dopo un'ora.

Il prezzo era fisso, 1 BNT = 0.01

ether.

Dall'analisi del contratto di vendita appare che in realtà il cap segreto non è esattamente segreto, è semplicemente inesistente (ovvero cap infinito) fino a che il team non decide di settarlo e dunque rivelarlo. Questo ha un significato diverso dal dire che il cap esiste ma è segreto.

Lo snippet di codice mostra il meccanismo esatto

```
400     /**
401         @dev enables the real cap defined on deployment
402
403         @param _cap    predefined cap
404         @param _key    key used to compute the cap hash
405     */
406     function enableRealCap(uint256 _cap, uint256 _key)
407         public
408         ownerOnly
409         active
410         between(startTime, endTime)
411         validEtherCap(_cap, _key)
412     {
413         require(_cap < totalEtherCap); // validate input
414         totalEtherCap = _cap;
415     }
```

Una caratteristica interessante è che un acquirente potrebbe portare il suo pagamento in priorità nella coda dei miner indicando un gas price più elevato. Questa non è una caratteristica

della ICO di Bancor, è una possibilità prevista da Ethereum. Per evitare un escalation del gas price, il team di Bancor ha imposto che tutti i pagamenti dovessero avvenire al costo fisso per il gas pari a 50 gwei.

ICO success

In questa sezione e nella successiva analizziamo due casi esemplari. Il caso di successo della ICO Bancor il caso di insuccesso della ICO InChain. Il confronto a breve distanza è utile per dare un'idea del quadro complessivo ma in generale si tratta di un'analisi a posteriori quindi nessuna delle

indicazioni date qui potrà garantire la ricetta giusta per la ICO vincente.

Abbiamo già conosciuto il progetto Bancor nella sezione dedicata. Il team viene dal mondo dell'internet media e il suo fondatore è stato anche il fondatore di Metacafe. Questo è un aspetto molto importante. Il fondatore di Bancor è anche il fondatore di una internet company che è stata sconfitta niente meno che da Youtube. Tuttavia Metacafe non è un nome sconosciuto, io stesso ho usato Metacafe prima che Youtube diventasse popolare. La prima osservazione è che il team ha una reputazione notevole dato che pur perdendo la battaglia del video online

l'ha persa contro il gigante e attuale incumbent di mercato. Nella mitologia internettiana anche le sconfitte contano.

Il fondatore di Bancor e di Metacafe racconta la genesi della sua idea più o meno in questo modo: Metacage ha perso contro Youtube perché si concentrava su contenuti di alta qualità mentre Youtube ha puntato sulla coda lunga dei contenuti "prodotti dal basso" ovvero il famoso user generated content. Milioni di video inutili e di scarsa qualità diciamolo pure. Nel caso del video pare che la coda lunga sia il 99% del totale, e solo l'1% sono le view su video professionali e featured.

Reduci da questa dura lezione il team di Bancor ha capito una cosa: anche nel mondo delle cripto bisogna puntare sulla coda lunga e non su Bitcoin. Quindi nasce Bancor come un sistema che consente a tutti di creare con poco sforzo un token e la relativa ICO, ma soprattutto di garantire a ogni token, anche il più sconosciuto, di poter essere quotato e a disposizione dei traders anche in assenza quasi totale di liquidità.

Veniamo alla ICO vera e propria. Ricordo che gli ingredienti fondamentali per la ICO sono il token ERC20 ed il white paper. Questo da un punto di vista puramente implementativo, in pratica

servirà anche tanto marketing e costruire un'immagine vincente e convincente.

In questo il team di Bancor si è mostrato molto valido. La copertura mediatica è stata molto buona, ci sono decine di video in rete e le persone chiave del progetto hanno partecipato e parlato a molti eventi importanti del mondo blockchain.

Non ultimo, i detrattori. Tanto più è autorevole il tuo detrattore tanto più il tuo progetto è meritevole di attenzione. Nel caso di Bancor ne abbiamo avuto alcuni importanti. Emin Gun Sirer, un famoso studioso di criptovalute, nel suo Bancor is Flawed non è andato molto

per il sottile.

http://hackingdistributed.com/2017/06/is-flawed/

E la successiva replica di Eyal Hertzog (founder di Bancor) https://blog.bancor.network/this-analysis-of-bancor-is-flawed-18ab8a000d43 è un chiaro esempio di come una critica possa servire in realtà a creare aspettativa e attenzione.

ICO failure

Le storie di successo sono in genere le uniche storie che sentiamo. Nessuno ci racconta o porta alla nostra attenzione le storie e i casi di insuccesso? Eppure come per la mitologia delle startup unicorno, sono tante le storie di progetti e delle loro ICO che non hanno raccolto la cifra sperata e che per una ragione o

per l'altra possono essere considerate dei fallimenti.

Perché una ICO fallisce? Possiamo analizzare alcuni casi esemplari e trarne le conclusioni o come dicono gli inglesi i caveats.

Inchain ad esempio era un progetto di assicurazione basata su blockchain. La sua value proposition in particolare era semplice e molto promettente, in pratica delle polizze assicurative basate sulla blockchain e cartolarizzate in forma di token.

Perché In Chain non ha raggiunto il suo obiettivo? Secondo il sito

https://medium.com/iconominet/why-icos-fail-1f9530a6d135 tra le ragioni annoverate c'è la scarsa "dimostrabilità" delle skill del team. Se il team non è in grado di mostrare delle skill sul progetto o sulla tecnologia blockchain questo è, come facilmente immaginabile, un grosso difetto. In particolare InChain non disponeva di una proof of concept solida, in pratica, un gruppo di sconosciuti stava chiedendo dei fondi per un concept tecnologico non dimostrato, senza un modello di monetizzazione convincente e senza proteggere gli investitori.

In effetti reperire informazioni sugli elementi del team non era immediato, tra

i founder Dmitry Lazarichev è presente su Linkedin e appare come un imprenditore senza tuttavia spiccare nelle doti relative alla ICO, ovvero quelle della blockchain come tecnologia e delle assicurazioni come dominio. Dell'altro founder Sergey Primatchik e del blockchain architect Andrey Zamovskiy Linkedin non mostra tracce. Naturalmente la loro assenza su Linkedin non significa una mancanza di competenze coì come un profilo su Linkedin non garantisce viceversa la capacità di ciò che viene riportato, tuttavia personalmente come investitore avrei sicuramente dato uno sguardo a Linkedin per ottenere qualche informazione in più e non trovare niente

mi avrebbe lanciato un immediato segnale negativo.

Un altro aspetto è legato probabilmente alla copertura mediatica del progetto. Il progetto è stato annunciato su bitcointalk come doveroso primo passo, si è dotato di un blog, ma a parte questo non sembra essere disponibile una documentazione di eventi e di video su youtube dove il progetto e la sua ICO sono stati presentati.

ICO: take-aways

Non esiste una ricetta per la ICO di successo. E' probabile che le ICO che hanno avuto successo nel 2016 non sarebbero in grado di ripetere il loro risultato oggi o fra un anno. E' tutto molto dinamico e inoltre aumenta la consapevolezza e la diffidenza di chi investe.

Il white paper è sicuramente un elemento essenziale, ma penso anche che chi investe per pura speculazione si basi sui rumors e il white paper non lo legga neanche.

Chi legge il white paper allora? Ma semplice! Chi vuole demolire la vostra ICO. Infatti saranno i vostri detrattori i primi a capire il vostro progetto. Per questo motivo il white paper deve essere chiaro e non deve promettere cose insostenibili.

La reputazione è tutto. Se siete noti negli ambienti che contano, se potete parlare bene in inglese nelle conferenze

che contano, se avete alle spalle una storia mitica di internet allora siete sulla buona strada. Se siete un pinco pallino qualunque della provincia brianzola (potrei dire sarda, rumena, serba, tunisina, basca e cosi via) la strada è più dura.

Alla fine la ricetta non esiste ma gli ingredienti sì:

- Un white paper inattaccabile
- Un token che dia il senso della scarsità e dell'urgenza di essere acquisito
- Tanto marketing, e tanta tensione intorno all'evento. Il countdown è un classico.

Test: valutare una ICO

Questa serie di domande è liberamente ispirata allo Joel Spolsky test che serve per autovalutare la propria software house. Questo è invece applicato alle ICO e le domande sono state elaborate a partire da diversi interventi ai quali ho assistito e post che ho letto di autori diversi tra cui alcuni di Giacomo Zucco, Ferdinando Ametrano, David Siegel e altri.

Disclaimer: questo test non garantisce nessun risultato. Una ICO

che superi brillantemente il test non implica un buon investimento e vice versa. In realtà ***non esiste una condizione per cui il test è superato, più domande hanno una risposta positiva più aumenta il livello di informazione e trasparenza*** *della ICO. Quindi non basarti solo su questo per fare i tuoi investimenti. Questo test è solo un modo per evidenziare alcune caratteristiche di una ICO.* ***Fai la tua ricerca.***

IL PROGETTO

Check fondatezza: il progetto

affronta un problema reale?

Check tecnologico: il problema anche se reale è risolvibile in modo pratico?

Check di mercato: la soluzione anche se praticabile ha anche un mercato che possa sostenerla?

Check legale: la soluzione anche se economicamente sostenibile, è legalmente praticabile?

IL TOKEN

Intanto analizza se promette dividendi e profitti e somiglia ad un bond o ad un'azione. In tal caso molto probabilmente è una security e non supererebbe l'Howey test (vedi sezione dedicata).

Check regolatorio: se è "come" una security, è stata chiesta l'autorizzazione per la raccolta di capitali?

Check (in)utilità: se non è una security, garantisce qualche diritto d'utilizzo di un prodotto o servizio che verrà sviluppato?

Check Ponzi: se nessun prodotto/servizio è previsto, possiamo

scongiurare che non sia uno schema piramidale dove chi entra prima ha dei vantaggi a scapito di chi entra nelle fasi successive?

Check (dis)economia: il token ha qualcosa di speciale nella dinamica e nella scarsità e non può essere rimpiazzato direttamente dalla valuta nativa (eth o btc)?

IL TEAM

Check anonymous: il team mostra le identità dei suoi membri?

Check skill: il team ha le

competenze per risolvere il problema?

Check organizzazione: il team ha le capacità organizzative per usare le risorse in modo efficiente?

Check liquidità: il team ha la reputazione e la notorietà per riuscire a listare il token sugli exchange?

Check "prendi i soldi e scappa": la giurisdizione territoriale in cui opera il team offre qualche garanzia a chi investe di vedere rispettati i propri diritti?

LA TOKEN SALE

Check codice sorgente: il contratto di token sale è open source?

Check vulnerabilità: è stato fatto un audit sugli smart contract della tokensale da uno o più esperti indipendenti?

Check FOMO: Fear of missing out, la token sale garantisce che in caso di transazioni appese, in coda e fuori tempo utile i miei fondi non vadano persi?

Check fallimento: la token sale ha un obiettivo minimo di raccolta noto e prevede la restituzione in caso di mancato raggiungimento?

Check superfinanziamento: la token sale ha un massimo di raccolta capitale noto?

Check termini legali: esiste un documento che spiega i termini legali della token sale?

Check automazione: la restituzione e altri termini della ICO sono codificati nello smart contract in blockchain o tramite escrow o altri automatismi?

Check coerenza: i termini legali sono corrispondenti a quanto decantato nel white paper?

Check (in)completamento lavori: esiste uno smart contract o equivalente sblocco dei fondi al raggiungimento dei risultati parziali ottenuti?

SPECULAZIONE

Check pump & dump: il team garantisce di non trattenere una percentuale sostanziale dei token?

Check manipolazione: il team garantisce di non poter bruciare o creare nuovi token a piacimento?

Check speculazione occulta: il sistema non prevede di bruciare token con l'utilizzo per aumentarne il valore?

Ramificazioni legali

Hai mai letto uno smart contract

Quando si investono dei soldi sarebbe sempre opportuno leggere il contratto che regola l'investimento. Nel caso degli investimenti tradizionali le banche o i proponenti devono in qualche modo informare i clienti di quali sono le conseguenze dell'investimento, qual è il

rischio di perdere il capitale eccetera. In altre parole devono assicurarsi che l'investitore sia consapevole del rischio che corre.

Questo meccanismo di trasparenza è spesso più teorico che pratico. Quante volte abbiamo sentito notizie di piccoli risparmiatori coinvolti in investimenti in titoli derivati, opzioni binarie o altre diavolerie finanziarie di cui non conoscevano assolutamente niente.

Nel mondo delle cripto le cose possono essere altrettanto rischiose o magari anche di più. In effetti non esiste l'obbligo di un prospetto informativo e anche se esistesse quello che conta è lo

smart contract stesso. Infatti a differenza di un contratto tradizionale, che in caso di vertenza potrebbe essere impugnato e portato all'attenzione di un giudice il quale a sua volta potrebbe interpretarlo a favore di una o dell'altra parte, un contratto su blockchain ha solo un'interpretazione ed è quella che scaturisce dall'esecuzione dello stesso su un computer. Così come è difficile a volte capire i termini di un contratto tradizionale, leggere e capire un contratto su blockchain richiede delle competenze ancora più specialistiche, non fosse altro che non è scritto in italiano (o in inglese) ma è scritto in un linguaggio di programmazione che in genere richiede delle competenze

acquisibili solo dopo anni di studio ed esercizio.

Per mitigare questo effetto di asimmetria informativa tra chi propone e chi investe, ultimamente molti progetti di una certa importanza hanno comunque fornito alcune garanzie operative, una su tutte dovrebbe essere quella che vede la restituzione delle somme se la raccolta non raggiunge l'obiettivo minimo stabilito.

Dal momento che chi propone una ICO lo fa attraverso una società incorporata allo scopo in qualche giurisdizione di vantaggio, oltre al white paper e allo smart contract della token

sale esiste quasi sempre un documento con i termini legali che regolano la vendita. Un rapido esame consente spesso di verificare che ciò che lascia intendere il white paper sembra molto più di quello che in realtà i termini legali stabiliscono.

Escrowed ICO

Abbiamo imparato che la ICO consiste nel raccogliere un certo quantitativo di coin ad un determinato indirizzo, sia questo Bitcoin o Ethereum o altro. Come abbiamo ormai appreso le transazioni verso un indirizzo sono irrevocabili, dunque cosa succede ai fondi versati dagli investitori se il team decide di

scappare con i soldi raccolti? Non esiste un metodo a posteriori per recuperarli e gli investitori sarebbero vittime dell'ennesimo scam.

Una possibilità è quella che il team della ICO non possa accedere immediatamente alle risorse raccolte ma possa farlo solo se un'entità terza e al di sopra delle parti garantisce. Questo non è altro che lo schema di escrow che abbiamo già discusso nella sezione dedicata ai contratti.

Non esiste un modo unico per implementare uno schema escrow. Ad esempio in Bitcoin si può fare con gli indirizzi multisig mentre in Ethereum

con un apposito contratto solidity.

Una ICO escrowed in generale garantisce maggiormente gli investitori e naturalmente potrebbe adottare anche schemi di **sblocco dei fondi a raggiungimento di obiettivi specifici**. Ad esempio se vengono raccolti 1000 ether, l'escrow potrebbe rilasciare immediatamente 300 ether per iniziare lo sviluppo, 400 ether al raggiungumento di una milestone come ad esempio la versione 1.0 della piattaforma, e il saldo di 300 eth dopo un mese di uptime. Il ruolo dell'escrow come arbitro e valutatore sarebbe centrale e molto importante.

Una caratteristica è quella che attraverso una ICO escrowed è possibile che il progetto si fermi e i fondi vengano restituiti ai rispettivi investitori. Come questo possa accadere dipende ancora una volta dall'implementazione.

Il caso della ICO InChain è esemplare in quanto la ICO aveva anche un limite minimo che non è stato raggiunto e per tale ragione il team ha deciso di rimborsare gli investitori. La ICO era appunto escrowed e accettava valuta in bitcoin e ether. A tal proposito ha nominato due diversi escrow, uno per il mondo BTC ed uno per ETH rispettivamente.

Vediamo qui alcuni dei passaggi più

interessanti.

Who am I?

Iam SebastianJu, Iam escrow on Bitcointalk since years. Iam not connected to the issuer or the coin. Iam an independent third party who acts as an escrow in an attempt to make the ICO more secure.

Safety from Escrow

The invested coins will be released to the issuer in the steps and milestones they provided as info.

Queste righe sopra spiegano esattamente la natura di una ICO escrowed

In generale poi per una corretta identificazione degli acquirenti dei token è importante che questi acquirenti spediscano i fondi da un indirizzo di cui controllano le chiavi privati. Quindi partecipare ad una ICO attraverso un exchange è fortemente sconsigliato.

Only send Bitcoins from an address you can sign a private message from or at least where you are able to get the private key for that address! This is important since in case of REFUND

you can't proof otherwise that you sent bitcoins when you, for example, sent the bitcoins directly from an Exchange.

Naturalmente l'escrow si fa pagare e soprattutto nel caso che poi debba occuparsi della restituzione dei fondi si riserva di trattenere una fee. Ecco il listino prezzi del caso in esame.

Minimum fee of 0.01 BTC or the fee structure, whatever is higher:

>=$500,001 = 0,5%
$500,000 - $50,001 = 1%
$50,000 - $10,001 = 2%

$10,000 - $1,001 = 3%
$1,000 - $101 => 5%
<=$100 => 10%

Howey test

La blockchain ha creato un nuovo territorio, una zona grigia e poco regolamentata. E' proprio per questa ragione che è un forte abilitatore di innovazione. Nel mondo "normale" se un'azienda emette bond o azioni deve seguire certe regole. Ci sono i compiti da fare, non è sufficiente mettere il

codice IBAN sul sito e un pdf con il business plan. Nel mondo cripto invece sembrerebbe tutto così facile. E tuttavia non è proprio così, per quanto riguarda gli USA esiste un test che la SEC, l'organismo di vigilanza sulla borsa, utilizza per stabilire se un'azienda sta eludendo i suoi controlli e le sue procedure. Questo test si chiama test di Howey e si applica anche alla raccolta di capitale in criptovaluta.

Il nome risale al giugno del 1946, quando la società agricola Howey decise di affittare metà della sua proprietà terriera per "finanziare un ulteriore sviluppo del suo business"

Il caso di Howey ha sfidato la convinzione prevalente di ciò che si intendeva quando si parlava di azioni e obbligazioni. Il contratto configurava di fatto un accordo tra un soggetto che forniva la gestione ed un altro soggetto che forniva il capitale (attraverso un contratto di locazione). Questo secondo soggetto "privo delle conoscenze, delle abilità e delle attrezzature necessarie per la cura e la coltivazione degli alberi di agrumi", diventava proprietario di terreni nominali grazie alla firma su un contratto.

Grazie a questo contratto di locazione questi investitori divennero effettivamente speculatori con l'aspettativa che avrebbero guadagnato

un profitto solo attraverso gli sforzi del promotore.

Mancando di registrare tali operazioni alla Securities Exchange Commission (SEC), Howey Co. aveva violato la legge federale.

La SEC presentò un provvedimento per impedire la vendita di questi contratti e nel maggio 1946 la Corte Suprema degli Stati Uniti ha confermato tale posizione affermando che i contratti di Howey erano contratti di investimento e come tali dovevano essere regolati.

Sembra incredibile che ancora oggi dopo 70 anni la sentenza di SEC v.

Howey possa avere un impatto sul mercato dei servizi legati alla blockchain.

In generale secondo tale test siamo in presenza di un contratto di investimento se:

1. E' presente un investimento in denaro o altro asset equivalente
2. Esiste una promessa di profitto atteso conseguente l'investimento
3. Chi investe si attende dei profitti che derivano dalla gestione dei fondi raccolti.

Il fatto che l'investimento possa essere considerato tale anche se erogato in asset equivalenti al denaro piuttosto

che usando denaro a corso legale rende le ICO delle operazioni soggette al test di Howey.

La SEC e theDAO

Come già introdotto in precedenza, la Security and Exchange Commission (SEC) è l'organismo di controllo che vigila sul rispetto delle leggi americane per quanto riguarda gli strumenti finanziari e la borsa. Naturalmente ogni decisione della SEC ha valore solo nel territorio americano, tuttavia è facile

immaginare che le sue conclusioni **possano ispirare gli equivalenti organismi di controllo** in altri paesi a prendere provvedimenti simili e avviare simili investigazioni.

La SEC ha analizzato il caso di TheDAO (vedi sezione *le DAO e TheDAO*) per determinare le responsabilità e inquadrarlo nel contesto delle regole e leggi vigenti. Il risultato di questa investigazione è un rapporto dettagliato che mette alcuni paletti e fa un po' di chiarezza rispetto all'idea di totale deregulation ispirata dalle nuove tecnologie.

Intanto il rapporto SEC è considerato

dagli esperti un piccolo capolavoro sia per quanto riguarda gli aspetti tecnologici che non sono mai banalizzati e anzi mostrano come i funzionari della Commissione sappiano il fatto loro districandosi nel gergo tecnico e utilizzando gli strumenti online per reperire dati e informazioni direttamente dalla blockchain. In un certo senso la descrizione di TheDAO sembra essere una delle più accurate tra quelle che si possono trovare in giro.

La primissima conclusione è che la Commissione non intende avviare una causa contro nessuno dei protagonisti come i proponenti, i curatori o gli advisor. Questo non significa che

qualcun altro non potrà in futuro fare causa a queste persone, ma il fatto che la SEC non lo faccia stabilisce sicuramente un fatto rilevante.

La seconda conclusione molto importante è che alla fine delle analisi svolte dalla Commissione questa ha stabilito che i Token di TheDAO sono security secondo la legge americana e come tali dovrebbero rispettarne i vincoli e le regolamentazioni.

Un aspetto che la Commissione ha voluto sottolineare nel rapporto è la asimmetria tra gli organizzatori e i curatori da una parte e i semplice token holder dall'altra. L'idea forte alla base

della DAO dovrebbe essere appunto quello che nessuno è in carica per decidere e la DAO è un'organizzazione puramente decentralizzata. Tuttavia data la natura dispersa, non coordinata, e anonima dei token holder, il gruppo di organizzatori e curatori aveva invece un grande potere decisionale. La mera capacità del token holder di votare è sicuramente un aspetto innovativo tuttavia non sufficiente a considerare curatori e organizzatori alla stessa stregua dei token holder. In pratica quello che la SEC mette in evidenza è che la DAO ha una struttura organizzativa non così decentralizzata in cui da una parte ci sono degli investitori che delegano gli organizzatori per

ottenere dei profitti dalle decisioni via via prese. Questo fa assomigliare la DAO ad una ben più tradizionale società per azioni in cui gli azionisti semplicemente attendono i risultati di esercizio dal consiglio di amministrazione. Per tale motivo la SEC considera i token di theDAO delle security.

Ultima conclusione degna di nota, il mercato secondario dei token è stato analizzato e la commissione ha stabilito che i vari operatori hanno operato all'interno delle regole previste. Sembra una buona notizia, ma in realtà non lo è per i fautori della non regolazione. Infatti sembra stabilire che il trading dei

token emessi da theDAO sia soggetto alle restrizioni e alle regole che la SEC stessa prevede per questo genere di strumenti. Quindi un monito per le piattaforme di trading che ora sono consapevoli che l'occhio della SEC è su di loro.

Ma quale può essere la conseguenza di tale rapporto sulle prossime ICO? Se i token della theDAO erano security lo sono anche tutti i token delle ICO che vediamo succedersi in questi giorni? Quali sono gli accorgimenti più semplici da prendere per evitare di finire con la testa sul ceppo?

Token mania

Vista la delicatezza del tema toccato dalla SEC e le conseguenze potenziali nell'emettere dei token crittografici che dovessero essere in un qualsiasi momento riconosciuti come strumenti di investimento, Coinbase, Coin Center, Union Square Ventures e Consensys nel dicembre del 2016 hanno predisposto un

vademecum su come distinguere i token di investimento da quelli funzionali che sbloccano dei diritti di utilizzo della piattaforma (A Securities Law Framework for Blockchain Tokens). Ricordiamo infatti che nel mercato americano oggi l'emissione di strumenti finanziari al di fuori delle regole previste dalla SEC è un reato. Il fatto di emetterli attraverso la blockchain non lo rende meno grave.

Possiamo individuare nei token di investimento le seguenti caratteristiche.

- Titolo di proprietà di un'entità giuridica, ad esempio un token

potrebbe corrispondere ad una frazione di equity.
- Diritto a utili
- Titolo di credito al portatore assimilabile a un bond.
- Facoltà di convertire il token in equity, credito o altro strumento finanziario regolamentato.

I token funzionali invece hanno le seguenti caratteristiche:

- Diritti di accedere a funzioni e risorse di calcolo della piattaforma
- Diritto di creare prodotti e ottenere royalties all'interno della piattaforma

- Diritto di voto sulla governance del sistema informatico (non della company che lo mantiene) e sulla sua evoluzione.

Terms and Conditions of the Ethereum Genesis Sale

E’ interessante analizzare alcuni dei passaggi delle Terms and Conditions della Ethereum Genesis Sale (from ethereum.org) gestita dalla società EthSwiss creata allo scopo di finanziare e sviluppare la piattaforma Ethereum.

Possiamo considerare Ethereum la madre di tutte le ICO, anche se non la prima in termini cronologici spettando questo primato alla piattaforma Omni. E' evidente che chi ha scritto questi termini voleva mettersi al sicuro da eventuali denunce e vertenze. Vengono riportati qui come cronaca dei fatti e **non sono assolutamente da intendersi come il modo corretto di corredare la vendita di token online o come un consiglio legale. Se vuoi andare sul sicuro non dovresti vendere token online. Se vuoi correre un ragionevole rischio rivolgiti ad un esperto legale esperto di nuove tecnologie, meglio se specializzato in questioni legali legate alla blockchain. Io non sono il tuo**

legale! Anzi, io non sono un legale, sono un ingegnere.

I termini di Ethereum partono con una negazione esplicita della natura di investimento azionario od obbligazionario. Anche se negarlo qui non significa che non lo sia e il test di Howey che per quanto riguarda la SEC rimane l'unico modo per stabilirlo.

This document is not a solicitation for investment and does not pertain in any way to an offering of securities in any jurisdiction

Per mettersi in un territorio ancora

più sicuro poi

Ownership of ETH carries no rights express or implied.

[...]
Purchases of ETH are non-refundable. Purchasers should have no expectation of influence over governance of the platform.

Riguardo poi la consapevolezza di chi acquista i termini indicano chiaramente che chi non è un esperto di criptovalute non dovrebbe acquistare gli ether.

WARNING: DO NOT PURCHASE ETH IF YOU ARE NOT AN EXPERT IN DEALING WITH CRYPTOGRAPHIC TOKENS AND BLOCKCHAIN-BASED SOFTWARE SYSTEMS

Riguardo gli aspetti cripto-economici i termini annunciano che non esiste un supply predeterminato dei coin distribuiti. Questo significa che a differenza dei bitcoin non c'è garanzia della effettiva scarsità.

EthSuisse will not place a cap on the number of ETH that can be purchased by the community.

Inoltre non spetta a EthSwiss verificare la liceità dell'acquisto. Questo controllo è accollato al compratore.

It is the responsibility of each potential Purchaser of ETH to determine if the Purchaser can legally purchase ETH in the Purchaser's jurisdiction.

Di seguito poi i termini sembrano voler rassicurare gli organismi di vigilanza stabilendo che chi compra non deve attendersi dei profitti dalla mera speculazione e che i coin sono funzionali all'utilizzo delle piattaforme

decentralizzate.

the Purchaser will take sole responsibility for any restrictions and risks associated with the purchase of ETH as set forth below; (v) represents and warrants that Purchaser is not exchanging bitcoin (BTC) for ETH for the purpose of speculative investment; (vi) represents and warrants that the Purchaser is acquiring ETH for the use of decentralized application services or the purchase of tokens specific to forthcoming decentralized applications

L'ultima parte dei termini mette in

evidenza tutta una serie di rischi che il compratore si accolla al momento dell'acquisto. Tra i vari rischi citati ad esempio c'è il rischio di volatilità di Bitcoin. Infatti dato che gli ether si comprano con i bitcoin, l'acquirente indica di comprendere che data la volatilità di questi il progetto Ethereum potrebbe dissolversi per mancanza di risorse.

[...]it is possible that the value of BTC will drop significantly in the future, depriving EthSuisse of sufficient resources to continue to operate[...]

Poi ovviamente il rischio legato alla

regolamentazione territoriale, chi compra i coin deve mettere in conto che lo sviluppo della piattaforma Ethereum potrebbe essere dichiarato illegale e bloccato prima del compimento.

[...] The Ethereum Platform and ETH could be impacted by one or more regulatory inquiries or regulatory action, which could impede or limit the ability of EthSuisse to continue to develop the Ethereum Platform[...]

Gli altri rischi rapidamenti enumerati sono:

Rischio di cloni:

...it is possible that alternative unofficial Ethereum-based networks could be established, which utilize the same open source source code and open source protocol [...] which could potentially negatively impact the Ethereum Platform and ETH.

Rischio di mancanza di interesse nel progetto.

[...]Such a lack of interest could impact the development of the Ethereum Platform and potential uses of ETH. EthSuisse [...] cannot predict the success of its own development

efforts or the efforts of other third parties.

Rischio di competitor nel campo delle applicazioni decentralizzate.

[…] **alternative platforms for decentralized applications may impact success of the Ethereum Project and the ability of EthSuisse to operate and sell ETH in the future**.

Rischio tecnologico ovvero di non riuscire a sviluppare Ethereum

Purchaser understands, […] that an official completed version of the Ethereum Platform may not be released and there may never be an operational

Ethereum Platform.

Rischio che EthSwiss si faccia rubare i soldi.

Hackers or other groups or organizations may attempt to steal the BTC revenue from the Genesis Sale, thus potentially impacting the ability of EthSuisse to develop the Ethereum Platform

E infine il disclaimer totale

23 Disclaimer of Warranties
THE PURCHASER EXPRESSLY AGREES THAT THE PURCHASER IS PURCHASING ETH AT THE

PURCHASER'S SOLE RISK AND THAT ETH IS PROVIDED ON AN "AS IS" BASIS WITHOUT WARRANTIES OF ANY KIND.

Aspetti legali: take-aways

Initial Coin Offering, in pratica fa il verso a Initial Public Offering, il processo con cui una società acquisisce capitali attraverso la vendita in borsa delle proprie azioni. La ICO avviene al di fuori dagli schemi normativi consueti, in pratica il team vende token in cambio di eth o btc. Cosa rappresentino i token è

tutto da capire. Ci sono varie opzioni:

- il token sblocca l'accesso ad un servizio/prodotto
- il token è equity o credito
- un mix delle prime due
- nessuna delle due

Nel secondo e terzo caso per la SEC/Consob sarebbe una security e quindi illegale se non fatta secondo i canoni previsti.

La domanda che bisogna porsi in pratica è se esiste un'aspettativa di profitto da parte di chi compra i token. Se la rispostà è sì, allora siamo nei guai.

Perché in molti ritengono le ICO uno scam senza se e senza ma?

- Perché uno strumento di finanziamento senza controlli permette gli abusi e quindi per molti è scam punto e basta.
- Perché stai cedendo un oggetto digitale a scarsità nota e sicurezza elevata (il btc) in cambio di un token a scarsità ignota e sicurezza non garantita.
- Perché per un vero bitcoiner vale !bitcoin == shitcoin -> True. Quindi è scam anche eth, figuriamoci un token preminato da me nel weekend.

Tuttavia non bisogna buttare il bambino con l'acqua sporca.

Ethereum è stato finanziato da una ICO, chi ha comprato gli eth a 1btc = 2000 eth oggi non si sente scammato ma miracolato.

Ci sono progetti under ICO molti interessanti, BAT per dirne uno. Le ICO sono una conseguenza inevitabile dell'esistenza delle cripto. Rappresentano le transazioni virtuali perfette, un oggetto digitale a scarsità in cambio di un altro.

Una regolamentazione pesante sarebbe contraria allo spirito stesso

delle cripto. In fondo preferisco essere libero di spendere i miei coin e perderli piuttosto che avere il regolatore che dice cosa posso o non posso fare.

Le ICO in questi giorni, meglio in questi istanti, stanno attraversando un'immensa fase di hype, non sottovalutiamo che però l'importanza nel lungo periodo: le forme di finanziamento attraverso contratti su blockchain e emissione di titoli crittografici saranno l'inevitabile futuro.

Criptoeconomia

Introduzione

Introduco questo capitolo con una domanda un po' provocatoria: **le criptomonete sono veramente monete?**

Forse no, forse sono non monete. Chi spenderebbe veramente un bitcoin per comprare qualcosa oggi? Io personalmente non lo farei se non fossi

costretto dalle circostanze. Per me la ragione è molto semplice, come posso desiderare di spendere oggi un bitcoin quando magari fra un mese il suo valore sarà raddoppiato. Il bitcoin è una moneta "deflazionistica" se mi passate il termine, non sono un esperto di teoria monetaria, ma questo rende per me i bitcoin praticamente inspendibili. Sinceramente io li considero un asset digitale da conservare per il futuro.

Naturalmente questa **è solo la mia opinione**. C'è chi sostiene che Bitcoin sarà un ottimo mezzo di pagamento in futuro e che la volatilità di oggi è dovuta alla fase espansiva attuale ma che quando sarà un sistema diffuso e usato

da tutti si smetterà di ragionare in termini di valore in dollari e semplicemente si ragionerà in bitcoin. Ho i miei dubbi su questo tuttavia nessuno può veramente dire l'ultima parola a proposito e solo il tempo ci dirà come andrà a finire.

D'altronde considerare i nostri bitcoin un mero asset su cui puntiamo per l'incremento di prezzo potrebbe portarci alla situazione della Greater Fool Theory, ovvero acquistare un qualcosa solo perché riteniamo che domani qualcun'altro vorrà acquistarlo a sua volta ad un prezzo più alto e così via all'infinito. Naturalmente in assenza di un vero valore sottostante il tutto si

riduce ad una clamorosa bolla speculativa. Tuttavia Bitcoin e le criptovalute in generale non possono essere solo questo per la grande quantità di innovazione che porta no con sé.

La moneta fiat sparirà prima o poi?

La moneta fiat sparirà? (ndr: fiat non sta per la marca di automobili)

Sembra improbabile. Ritengo non succederà finché esisteranno gli stati nazionali.

Il punto è come sempre quello **della legge del più forte**. La blockchain, i suoi contratti le sue monete hanno una grande bellezza e purezza che deriva dalla matematica ma purtroppo il mondo non è solo matematico ma è anche fisico. E' fatto di terra, di cemento, di ferro di sudore e sangue e c'è qualcosa che nella blockchain non possiamo transare. **La realtà fisica delle cose**.

Provo a spiegarmi meglio. Supponiamo di comprare una casa in bitcoin, il pagamento lo possiamo fare sulla blockchain ma la transazione in bitcoin è solo metà della transazione, l'altra metà, quella mancante, è il trasferimento di proprietà. Mentre il

pagamento avviene in modo decentralizzato, trustless, crittografico, immutabile, sicuro e soprattutto automatico e senza l'intervento dell'uomo, il trasferimento della proprietà e la garanzia che questa proprietà sia riconosciuta richiede che un'autorità ponga il suo sigillo. Questo si manifesta attraverso il notaio, il catasto, la giurisdizione del territorio, e in altre parole lo stato e la sua forza, che alla fine è sempre una forza fisica e militare. Se **non paghi il mutuo ti pignorano la casa**, se non lasci la casa ti portano via con la forza, la forza fisica che si misura in strattoni e joule.

Gli asset fisici sono sottoposti alla

forza fisica, gli asset digitali sono sottoposti alla crittografia, purtroppo però non possiamo vivere di soli oggetti digitali. Sarebbe bello? Oppure magari no, sarebbe orribile. Questo nostro essere "esseri" fisici ha anche qualche vantaggio: gustare una pizza, fare sesso o fare il solletico, immergersi in una jacuzzi, fare un giro in moto sulla strada costiera. Ma questa è pura filosofia. Il problema del corpo fisico è che abbiamo bisogno di cibo, riparo, vestiti e varie altre cose che ci possono essere sottratte o garantite da una forza fisica. La nostra forza personale nel caso della jungla, o la forza dello stato nel caso della società moderna.

E da qui alla moneta fiat il passo è breve. Vi immaginate di vivere **senza uno stato padre padrone e madre che allatta allo stesso tempo**. Che il più forte siate voi, il vostro vicino, o lo stato perché mai il più forte dovrebbe lasciare ad un sistema decentralizzato la gestione dello strumento tutto sommato più importante e più efficace per l'esercizio del potere, ovvero il denaro?

Non succederà mai, io credo che gli stati non lo permetteranno mai e i cittadini non faranno la rivoluzione anarchica per usare i bitcoin ed eliminare i dollari e gli euro.

Alla gente comune, concetto orribile

forse, non gliene importa nulla della centralizzazione e della decentralizzazione. Se gliene importasse davvero qualcosa invece di Facebook continueremmo ad usare NNTP e IRC (non sai cosa sono? Meglio, vuol dire che sei più giovane di me)

Se non ci vogliamo rassegnare allora dobbiamo immaginare che insieme alle monete fiat spariscano gli stati stessi che le emettono. **E' questo uno scenario possibile?** Possiamo immaginare una società senza stati nazionali e senza le loro monete? Un mondo della sovranità individuale? Ma cosa significa nella realtà? Visto che l'individuo tende ad essere avido e tende a sottomettere i

suoi simili per ottenere un vantaggio materiale, pare inevitabile la formazione di gruppi organizzati (l'unione fa la forza) e di centralizzazione (il più forte prende tutto). Oppure un mondo dove al posto degli stati ci sono le grandi multinazionali, dove Apple e Google battono moneta e hanno forze militari e i loro tribunali. Che mondo sarebbe? Syndicate (famoso videogame anni 90). Abbandoniamo immediatamente questa distopia, bisogna solo crescere e capire che la moneta fiat sarà il suolo su cui la moneta cripto prospererà: 33% hype, 33% speculazione, 33% innovazione.

Il paragone con le dotcom

Tra le possibili congetture sulla natura delle criptomonete c'è quella che le associa alle società dotcom dei primi anni duemila.

Le criptomonete sono le nuove dotcom?

Come sappiamo la storia quando si ripete lo fa con schemi e attori a prima vista irriconoscibili. Saremmo portati a pensare che le prossime dotcom debbano essere delle aziende registrate in qualche camera di commercio e non delle community. E invece forse, l'aspetto rivoluzionario sta proprio qui.

Partiamo dalla tecnologia. Così come il web negli anni 90 era nient'altro che un protocollo facile, decentralizzato ed efficace per trasferire l'informazione fra i computer, il protocollo che governa la blockchain è esso stesso un modo decentralizzato ed efficace per creare e trasferire un'attestazione notarile attraverso la rete ma senza un notaio. Da

qui la creazione di moneta, o forse l'equivoco della creazione di moneta.

Satoshi Nakamoto ha definito il concetto, l'ha creato e poi ha regalato l'embrione che si è successivamente sviluppato all'interno della comunità Bitcoin che come spesso accade nel mondo del software assume i connotati dei movimenti religiosi, con correnti più o meno fondamentaliste ed altre più tolleranti. I brani del paper di Satoshi vengono a volte citati come verità incontrovertibili, una specie di vangelo, una reductio ad satoshium (come la hitlerum ma al contrario) che dovrebbe mettere a tacere le dispute.

Vitalik Buterin è colui che ha sfidato il Verbo bitcoiniano con la creazione di Ethereum. Se con Bitcoin non era ancora chiaro che le cripto sono le nuove dotcom, Ethereum fornisce i primi segnali di questa trasformazione del concetto di comunità open source in cripto azienda. Ethereum non vuole essere una startup ma promette una piattaforma di calcolo decentralizzato e si fa finanziare attraverso i bitcoin. Vitalik e soci vendono con la cosiddetta presale le quote della nuova società (o cripto azienda) che si chiamano ether, dicendo appunto che questi sarebbero stati la benzina della nuova piattaforma. La moneta virtuale per finanziare l'esecuzione delle applicazioni

decentralizzate. Abbiamo già visto come i token possono essere di tipo finanziario (equivalenti a azioni o bond) oppure funzionali (che sbloccano qualche funzionalità nel sistema o assimilabili alla pre-vendita di un servizio). Nel caso di Ethereum come possiamo ignorare il fatto che gli ether abbiano una grande valenza da un punto di vista speculativo. Sappiamo che Ethereum non è un'azienda e quindi non possono essere azioni da un punto di vista formale. Tuttavia se estendiamo il concetto di azienda alla community e ai suoi attori principali, il cosiddetto ecosistema, non possiamo forse assimilare gli ether alle azioni della non-azienda Ethereum?

Possiamo forse spingerci a dire che ogni altcoin (alternative coin) è in realtà una nuova (non) azienda? In alcuni casi viene fondata attraverso una ICO o crowdsale come Ethereum, Synereo, Melonport mentre in altri no, è semplicemente uno sforzo di una team che poi riesce a ottenere una massa critica di utenti e poi vengono quotati in qualche grande exchange. Ci sono centinaia di criptovalute. Se si mettono in ordine di capitalizzazione Bitcoin risulta più grande della somma di tutte le altre messe insieme. Ma questo è normale. E' la power law, una legge naturale che regola molti aspetti della nostra vita che riprenderemo in seguito.

E’ un mercato in cui l’incumbent è Bitcoin e gli altri devono trovare i loro spazi, questi spazi sono pochi e le altcoin devono trovare una value proposition convincente.

Quando guardiamo la pagina di coinmarketcap.com stiamo assistendo forse ad listino azionario naturale di un nuovo settore industriale, senza autorità di controllo e che risponde direttamente alle forze naturali del mercato? La cosa interessante è che chiunque può acquisire le quote di queste nuove società comprando direttamente i loro coin.

Cosa producono le cripto aziende

Cosa producono queste cripto aziende?

Abbiamo già visto che le criptomonete e gli asset digitali non funzionano bene come monete almeno per ora. Sembra impossibile che i

bitcoin sostituiscano i dollari o gli euro visto che sono by design deflazionistici e nessuno li vuole spendere davvero. Magari accumulare sì, ma non certo spenderli per comprare una pizza come la famosa pizza di Laszlo pagata 10000 bitcoin all'alba dei tempi (2010) e che oggi risulterebbe costare oltre $40M. Laszlo non era certo uno stupido, all'epoca era necessario fare questo scambio per stabilire un prezzo diverso da zero. Il pizzaiolo diede al bitcoin la sua prima quotazione di mercato, 1/10000 di pizza.

Quindi cosa producono le cripto aziende? Domanda non banale. Tutte producono delle verità crittografiche

cristallizzate dentro blockchain più o meno immutabili. Bitcoin si configura come l'oro digitale e riserva di valore, Ethereum come la macchina incensurabile per l'esecuzione di denaro programmabile, Zcash come il mezzo di pagamento totalmente privato e così via le altre ognuna con la sua value proposition più o meno forte, o più o meno debole.

Bitcoin e i suoi piccoli fratelli quindi non sono dei protocolli informatici, o meglio non sono **solo** dei protocolli informatici come lo è ad esempio HTTP, sono invece dei sistemi in equilibrio economico analizzabili attraverso la teoria dei giochi.

Mantenere questo equilibrio costa, in particolare nel caso di Bitcoin costa energia elettrica che in modo più o meno indiretto è convertita in integrità del sistema. Senza l'incentivo economico sparisce l'integrità, sparisce l'utilità, sparisce la blockchain. Lo ripetiamo: **non c'è blockchain senza coin e non c'è coin senza blockchain.**

le DAO e theDAO

Se una macchina deve essere infallibile non potrà essere anche intelligente (A. Turing)

Speriamo! (il sottoscritto)

Abbiamo citato theDAO nella

sezione dedicata alla SEC e alle sue indagini. In questa sezione vediamo di descrivere meglio il concetto di Decentralized Autonomous Organization (DAO appunto) raccontando la storia di uno dei più grandi attacchi informatici che ha messo a rischio un patrimonio di oltre $150M. E' una storia che tutti dovrebbero conoscere, anche se non siete esperti di Bitcoin, perché probabilmente finirà nei libri di storia, o almeno su quelli di informatica.

Una DAO è un contratto o programma in esecuzione sulla blockchain con una dotazione finanziaria e che emette quote (token) che possono essere comprate con ether (la moneta di

Ethereum).

La DAO può ricevere proposte progettuali e finanziarle usando la sua dotazione. Idealmente un nuovo modello di azienda senza uffici, senza CEO/CTO/CIO eccetera che svolge in modo autonomo la gestione dei fondi e dove gli shareholder votano con una transazione sulla blockchain, più token hai, più il tuo voto conta.

Un'organizzazione così potrebbe in teoria essere scorporata da qualsiasi giurisdizione territoriale. Potrebbe essere la società di diritto crittografico.

TheDAO è stata l'incarnazione principale di questa idea ed in un certo senso il Titanic di tutte le DAO. La

posizione dei developer era più o meno questa “il codice è legge noi non abbiamo responsabilità”.

The terms of The DAO Creation are set forth in the smart contract code existing on the Ethereum blockchain at 0xbb9bc244d798123fde783fcc1c72d3bb Nothing in this explanation of terms or in any other document or communication may modify or add any additional obligations or guarantees beyond those set forth in The DAO’s code.

Sembrava un progetto eccitante, innocuo e senza drammatiche

conseguenze se non si fosse rivelato il più grande crowdfounding di sempre, almeno fino a Giugno 2016, raccogliendo 150 milioni di dollari in un mese.

Personalmente ricordo bene quelle giornate. TheDAO stava per chiudere la vendita dei suoi token e io mi trovavo in compagnia dei colleghi nella sala mensa di Intel a Leixlip quando sento uno di loro dire – Davide just invested 50 grands on TheDAO – io quasi mi strozzo col boccone – nah, just few hundreds, I like playing with new toys. Per fortuna avevo investito solo qualche centinaio di euro dato che il 17 Giugno TheDAO veniva violata. Un hacker aveva trovato

un meccanismo o meglio una vulnerabilità da sfruttare per prelevare tutti i capitali raccolti. Il processo di "sifonamento" non poteva avvenire subito, ma richiedeva da programma 27 giorni di tempo. 27 giorni in cui tutti gli interessati hanno cercato una soluzione per evitare il peggio.

Come ha fatto il DAO (o la DAO?) ad essere violato?

Il meccanismo individuato dall'hacker è questo: quando un partecipante vuole uscire dalla DAO può scrivere del codice che invoca una funzione del contratto chiamata splitDAO ( ) ma ci sono due problemi.

Il primo è che questa funzione

quando viene invocata prima di tutto SPEDISCE gli ether al richiedente e solo DOPO impone che il nuovo balance sia ZERO. Il secondo problema è che questa funzione poteva essere invocata più e più volte.

Vitalik Buterin propose un soft fork, ovvero una modifica delle regole del protocollo che aggiunge una restrizione. In pratica senza rendere "nulla" o invalidare nessuna delle transazioni già presenti nella blockchain ma aggiungendo una specie di blacklist delle transazioni aventi lo scopo di prelevare dall'account 0x727..a4ba (l'indirizzo della theDAO figlia). Questo avrebbe impedito all'hacker l'accesso ai fondi superata la

finestra temporale dei 27 giorni.

Ai miners non restava che stare calmi, far funzionare le transazioni come al solito e aspettare una versione del software da scaricare e installare se avessero condiviso questa scelta.

L'hacker però non gradì l'idea e scrisse al mondo – vi denuncio, non sono un ladro. Ma il codice non era legge? In effetti, una legge ingiusta in questo caso, ingiusta per chi?

A soft or hard fork would amount to seizure of my legitimate and rightful ether, claimed legally through the terms of a smart contract. (...)

I reserve all rights to take any and all legal action against any accomplices of illegitimate theft, freezing, or seizure of my legitimate ether, and am actively working with my law firm. Those accomplices will be receiving Cease and Desist notices in the mail shortly.

(TheDAO Hacker)

Non è incredibile? Un hacker che minaccia vie legali.

Ma visto che come minaccia legale era evidentemente poco credibile l'hacker trovò un modo forse più efficace. Tentò di corrompere i miner annunciando - votate per me. Che significa - **non adottate il soft fork**.

[S]oon we will have a smart contract to reward miners who oppose the soft fork and mines the transaction. 1 million ether + 100 btc will be shared with miners.

TheDAO Hacker

Ma l'hacker non ebbe bisogno di scongiurare il soft fork, qualcosa era sfuggito ai geni della Ethereum Foundation, qualcosa era sfuggito a Vitalik. Il soft fork non poteva funzionare.

In pratica il soft fork sarebbe diventato un vettore per un grande DoS (Denial of Service) che poteva essere

architettato più o meno così: facciamo un flooding della rete con transazioni computazionalmente pesanti, e poi come ciliegina finale un'operazione sul DAO contract (quello blacklisted). Ecco che i miner si trovano impegnati a minare delle transazioni che poi si trovano costretti a rifiutare, bruciando cpu senza guadagnare nessuna fee.

Gotta find a way. A better way (Nirvana)

Allora non restava che la soluzione drastica, quella che avrebbe diviso la community per sempre. In sostanza tornare indietro nel tempo e cambiare ciò che era scritto in blockchain.

In realtà si tratta di un ri-allocamento dei fondi sifonati dentro un nuovo contratto DAO, quest'ultimo "aggiustato" e che permette solo ai legittimi token owner di riprendersi i fondi depositati.

Ma è stato giusto o no fare questo salvataggio? In fondo da un lato abbiamo una blockchain in cui un solo individuo ha sifonato una buona parte degli ether in circolazione, creando una centralizzazione in termini di capitale, e creando un'enorme contraccolpo psicologico a tutti quelli che avevano creduto in Ethereum prima e in TheDAO poi. L'altro scenario, quello risultante dall'hard fork, consola i poveri

derubati, ristabilisce la giustizia, punisce il colpevole o per lo meno lo mette fuori gioco. Il giudizio non è così semplice, si è scelta la seconda strada ma si è creato un precedente nella tecnologia blockchain. Non è più vero che una transazione è per sempre se attraverso un'opera di discussione e persuasione si può revocare l'irrevocabile. Si compromette alla radice il concetto stesso di blockchain. In fondo se **hai puntato i tuoi soldi su theDAO hai sbagliato tu**. Non eri obbligato.

Al blocco 1920000 il fork è tuttavia riuscito, Ethereum è salvo, TheDAO è salva, o meglio inutile. Anch'io appena

possibile mi sono informato su come convertire i miei token in ether e dimenticare questa storia. E' stata una piccola esperienza negativa per il sottoscritto, immagino che qualcuno ci avesse investito parecchi soldi. Tutto sembrava tornare alla normalità. I blocchi si susseguivano al ritmo di uno ogni 15 secondi, come al solito, 1920001, 1920002, 1920003, … la vita ripartiva, in quel mondo ai più sconosciuto che è la blockchain (Ethereum).

Qui Blocco 1920000, sono Ethereum classic, vi parlo dal passato e non mi arrendo

Idealmente una Blockchain ha una sola testa che avanza nel tempo blocco dopo blocco come un serpente digitale che guarda sempre avanti. Se per qualche malaugurato caso la rete dei miner si divide e crea una seconda testa la Blockchain si divide, una parte della rete cresce sulla testa numero uno, e l'altra parte sulla testa numero due. In generale quando questo succede tutti i nodi se ne accorgono e il protocollo stabilisce cosa fare: eliminare la branca che risulta più "corta", diciamo con meno blocchi (non è così semplice, ma lasciamo stare i dettagli). In questo modo la branca perdente sparisce e la chain torna ad avere una sola testa. Questo vale quando la community è

coesa e il fork avviene per caso, ma cosa succede quando il fork è causato da un cambiamento delle regole e qualcuno non vuole invece aderire al cambiamento. Succede che nasce Ethereum Classic.

It is however, with deep regret, that we as a community have had to spontaneously organize to defend the Ethereum blockchain platform from its founding members and organization due to a long train of abuses, specifically by the leadership of the Ethereum Foundation.

(Ethereum Classic Declaration of Indipendence)

Questa subcommunity continuò a minare a partire dal blocco del fork, il 1920000 e da lì in su ma non con le nuove regole, bensì tenendo le vecchie, dove esiste una DAO sifonata ed un signor hacker che con il suo address prima o poi potrà intascare il maltolto.

Succedeva nel 2016, dopo oltre un anno, a Settembre 2017, Ethereum ed Ethereum Classic sono ancora tra noi più forti che mai.

Appendice

King.sol ABI

[{"constant":true,"inputs":
[],"name":"name","outputs":
[{"name":"","type":"string"}],"payable":
{"constant":true,"inputs":
[],"name":"price","outputs":
[{"name":"","type":"uint256"}],"payable
{"constant":true,"inputs":

[],"name":"king","outputs":
[{"name":"","type":"address"}],"payable
{"constant":false,"inputs":
[{"name":"_name","type":"string"}],"nan
[],"payable":true,"stateMutability":"paya

Codice di King.sol

```
pragma solidity ^0.4.16;
    /**
     * This is inspired by King of Ether
     */
    contract King{
        address public king=address(0);
        uint public price=0.001 ether;
        string public name="";
```

```
        function claim(string _name)
payable {
            var former = king;
            if(msg.value >= price){
                king = msg.sender;
                if(former!=address(0))
former.send(price);
                price = price + price / 10;
                name  = _name;
            }else throw;

        }

    }
```

ERC20 Token

```
// Abstract contract for the full ERC 20
Token standard
    //
https://github.com/ethereum/EIPs/issue

    contract Token {
        /* This is a slight change to the
ERC20 base standard.
```

function totalSupply() constant returns (uint256 supply);

is replaced with:

uint256 public totalSupply;

This automatically creates a getter function

for the totalSupply.

This is moved to the base contract since

public getter functions are not

currently recognised as an implementation of the

matching abstract

function by the compiler.

**/*

/// total amount of tokens

uint256 public totalSupply;

```
        /// @param _owner The address
from
        /// which the balance will be
retrieved
        /// @return The balance
        function balanceOf(address
_owner)
        constant returns (uint256
balance);

        /// @notice send `_value` token to
`_to` from `msg.sender`
        /// @param _to The address of
the recipient
        /// @param _value The amount of
token to be transferred
        /// @return Whether the transfer
was successful or not
```

```
    function transfer(address _to,
uint256 _value)
    returns (bool success);

    /// @notice send `_value` token
to
    /// `_to` from `_from` on the
condition
    /// it is approved by `_from`
    /// @param _from The address of
the sender
    /// @param _to The address of
the recipient
    /// @param _value The amount of
token to be transferred
    /// @return Whether the transfer
was successful or not
    function transferFrom(address
```

```
_from,
                    address _to,
                    uint256 _value)
returns (bool success);

    /// @notice `msg.sender`
approves `_addr` to spend
    /// `_value` tokens
    /// @param _spender The address
of the account
    /// able to transfer the tokens
    /// @param _value The amount of
wei to be approved
    /// for transfer
    /// @return Whether the approval
was successful or not

    function approve(address
```

```
_spender,
                    uint256 _value)
        returns (bool success);

        /// @param _owner The address
of the owner
        /// @param _spender The address
of the account
        /// able to transfer the tokens
        /// @return Amount of remaining
tokens allowed to spent
        function allowance(address
_owner, address _spender)
        constant returns (uint256
remaining);

        event Transfer(address indexed
_from,
```

```
                address indexed _to,
                uint256 _value);

        event Approval(address indexed
_owner,
                address indexed
_spender,
                uint256 _value);
    }
```