UNIVERSITÀ DEGLI STUDI DI GENOVA

Dipartimento di Ingegneria meccanica, energetica, gestionale e dei trasporti

Dottorato di ricerca in Ingegneria delle macchine e dei sistemi per l'energia, l'ambiente e i trasporti

Curriculum Matematica e Simulazione

XXX ciclo

Tesi di dottorato

Algoritmi innovativi a supporto di modelli di simulazione

Relatore: Professor Bruzzone Agostino

Candidato: Valentina Bertella

Indice

Introduzione

Il tema principale della tesi è lo studio di algoritmi a supporto della simulazione.

La motivazione principale che ha mosso questo studio è stata quella di capire se e come la tecnologia blockchain possa essere utile in simulazione.

Un primo sforzo fatto è stato quindi quello di comprendere e dare una descrizione della tecnologia blockchain indipendentemente dalle sue applicazioni finanziarie: anche se è ben nota la sua potenzialità al di là del mondo delle criptovalute, a partire dalla sua introduzione in Bitcoin ([Nakamoto, 2008]), i report al riguardo sono rari e spesso generici.

Una blockchain può essere definita come un database distribuito che, sfuttando la tenologia peer to peer, gestisce un elenco sempre crescente di dati, mediante una procedura di codifica a blocchi (elemento caratterizzante della

tecnologia), ideata per dare a tali dati sicurezza e immutabilità, mediante l'uso di strumenti crittografici e di un sistema efficace di verifica. Nella Sezione 1.1 viene data una breve descrizione della tecnologia e vengono classificate le varie tipologie di blockchain in base ad opportuni criteri: rispetto alla loro applicazione ([Swan, 2015]) e al permesso (necessario o non) di partecipare alla rete ([Buterin, 2015]).

In un contesto generale, una transazione può essere definita come un'operazione su un oggetto della blockchain; relativamente ad essa, avremo un mittente e un destinatario i quali, come tutti gli utenti della blockchain, sono identificati da un indirizzo pubblico (generato dalla propria chiave pubblica), mentre la firma digitale che apporrano sulla transazione sarà generata da una sign function mediante la propria chiave privata (assegnata in input alla sign function stessa).

Dopodiché la transazione viene trasmessa in rete, dove avviene la verifica indipendente della sua validità da parte di ogni nodo validatore. Speciali nodi aggregano tutte le transazioni valide, ma non confermate, in un blocco candidato. A questo punto è necessario un accordo (consenso) tra i nodi su quale blocco deve essere aggiunto alla catena: viene quindi applicato un *consensus algorithm*, necessario per risolvere il problema della sincronizzazione

in un database distribuito.

Un secondo sforzo fatto è stato quello di studiare e descrivere gli algoritmi di consenso noti ed offrire una panoramica descrittiva dello stato dell'arte su tale argomento, non nota al momento della scrittura di questa tesi (Sezione 1.2), oltre a fornire un'analisi di confronto tra gli algoritmi di consenso stessi. Come sopra detto, questi ultimi sono lo strumento attraverso il quale i nodi della rete si accordano su quale blocco aggiungere alla catena. A seconda di come tale accordo viene raggiunto, essi sono stati da noi suddivisi in due grandi categorie: *proof-based* e *vote-based* . La scelta di tale dicitura è stata dettata dal fatto che nel primo gruppo il blocco sarà aggiunto alla catena dal nodo che avrà *dimostrato* in qualche modo (differente nei vari algoritmi) di essere il più qualificato degli altri nel farlo; invece nel secondo gruppo un blocco sarà aggiunto solo dopo che i nodi (o alcuni di essi, a seconda dello specifico protocollo) si saranno *scambiati opinioni* riguardanti i propri risultati di verifica indipendente.

Nella Sottosezione 1.2.1 viene descritto l'algoritmo *Proof of work*, algoritmo proof-based ideato da Adam Back ([Back, 1997]) e, come ben noto, utilizzato in Bitcoin per il processo di mining; unitamente alla descrizione, viene evi-

denziata la percentuale di potenza avversaria tollerata e vengono sottolineati i lati negativi.

Vengono poi descritte le varianti di Proof of Work, nate ciascuna per ovviare ai lati negativi di cui sopra: in [Eyal et al., 2014, Miller et al., 2015] troviamo proposte per prevenire la formazione di mining pool; in [Tromp, 2014], nella *Proof of Space* ([Dziembowski et al., 2013]) e nella *Proof of Burn* ([Stewart, 2012]) troviamo tre varianti per porre rimedio al problema dei costi elevati di elettricità e di hardware, mentre in [Eyal et al., 2016] viene proposto l'utilizzo di micro-blocchi per permettere una verifica più veloce delle transazioni; la variante Greedy Heaviest-Observed Sub-Tree ([Ghost, 2015]) è nata per cercare di trovare un compromesso tra velocità e sicurezza, e differisce dalla Proof of Work nel comportamento di fronte ad un fork; gli algoritmi *Proof of eXercise* ([Shoker, 2017]) e *Proof of Useful Work* ([Ball et al., 2017]), sono nati con lo scopo di ottenere output utili per risolvere problemi pratici ed evitare sprechi di risorse. Altri algoritmi di tipo proof-based sono la *Proof of Elapsed Time* [Rilee, 2018], la *Proof of Retrievability* ([Bowers et al., 2009]) e la *Proof of Luck* ([Milutivonic et al., 2016]).

Anche l'algoritmo *Proof of Stake*, algoritmo di tipo proof-based la cui prima proposta risale al 2011 ([bitcoin forum, 2011]), nasce per ovviare al problema

della Proof of Work legato al dispendio energetico e alla necessità di possedere hardware specializzati: con la Proof of Stake, la probabilità di aggiungere un blocco alla blockchain è proporzionale all'interesse (*stake*) e non più alla potenza computazionale. Oltre al risparmio energetico, l'idea di base è quella per cui chi possiede più ricchezza dovrebbe essere più fidato e, di conseguenza, dovrebbe essere meno probabile un suo attacco alla rete.

Ovviamente non solo la quantità di interesse può incidere sulla scelta del nodo (*leader*) che aggiungerà il blocco (altrimenti colui che possiede più stake sarebbe l'unico a costruire la catena), quindi in generale nella Proof of Stake è presente anche un altro fattore che dipende dalle varie versioni e che principalmente è stocastico.

Nella Sottosezione 1.2.2 viene descritto l'algoritmo Proof of Stake utilizzato in Nextcoin e vengono poi analizzate alcune sue varianti, che differiscono proprio nella randomizzazione che porta all'elezione del leader. Dapprima viene descritto il protocollo *Chains of Activity* (CoA), proposto in [Benton et al., 2016], che utilizza la *follow-the-satoshi procedure*, già descritta in [Benton et al., 2014]. Gli autori descrivono il protocollo Chains of Activity come il più sicuro degli esistenti protocolli rispetto ad alcuni attacchi, ma in [Kiayias et al., 2017] gli autori obiettano il fatto che in [Benton et al., 2016]

non viene fornito un modello formale per l'analisi dei protocolli Proof of Stake e che le prove di sicurezza non si basano su precise definizioni; quindi, in [Kiayias et al., 2017], gli autori forniscono un modello generale di costruzione di protocolli di tipo Proof of Stake, propongono il nuovo protocollo *Ouroboros*, e forniscono argomentazioni formali (che coinvolgono probabilità, combinatoria e teoria dei giochi) per dimostrare la sicurezza del protocollo proposto. In particolare viene ideato un nuovo meccanismo di incentivazione alla partecipazione al protocollo che viene dimostrato essere un equilibrio di Nash; inoltre il tipo di randomizzazione dell'elezione del leader è costruita con lo scopo di renderla il meno esposta possibile alla *grinding vulnerability*, ovvero alla manipolazione da parte di avversari che, simulando il protocollo, potrebbero predire il calcolo e quindi condizionare l'elezione.

Il *delegated proof of stake* ([Larimer, 2014]) propone una variante all'algoritmo di Proof of Stake con lo scopo di accelerare i tempi di conferma delle transazioni e di validazione dei blocchi, in quanto i nodi addetti a tale compito sono minori: gli stakeholder eleggono una commissione di delegati che avranno il compito di verificare le transazioni, generare e validare i blocchi ed inserirli nella catena.

Nella sottosezione 1.2.3 vengono descritti i protocolli di tipo vote-based, par-

tendo da protocolli di tipo *Bizantine Fault Tolerance*. Nelle blockchain che utilizzano tali algoritmi di consenso, le transazioni ed i blocchi devono essere verificati da tutti i nodi validatori insieme, in modo da avere univocità nel consenso; tali nodi devono dunque essere noti per poter comunicare tra loro prima di decidere se aggiungere un blocco alla catena o meno e, per questo, vengono per lo più adottati nelle blockchain di tipo permissioned. I protocolli di tipo Bizantine Fault Tolerance possono essere suddivisi in due categorie: *broadcast-based* e *chain-based* ([Duan et al., 2014]); la principale differenza sta nelle performance, in quanto i protocolli appartenenti alla seconda categoria mirano a raggiungere flussi più elevati a spese di una maggiore latenza; tuttavia, al crescere del numero dei nodi, i protocolli di tipo chain-based possono ottenere latenza minore rispetto a quelli di tipo broadcast-based. Per contro, i protocolli di tipo chain-based sono meno resilienti ai guasti. Vengono quindi descritti il principale esponente di protocollo broadcast based (il *Pratical Byzantine Fault Tolerance*, introdotto in [Castro et al., 2002]), la sua variante *Tendermint core*, ed il principale esponente di protocollo chain-based (*BChain*, [Duan et al., 2014]), che adotta l'approccio chiamato *re-chaining*, che consiste in un'operazione a basso dispendio computazionale mediante la quale la catena viene riordinata nel caso di sospetto di guasto, con lo sco-

po di migliorare la resilienza ai guasti. L'algoritmo di consenso *Sumeragi*, implementato in Hyperledger Iroha, trae ispirazione dal protocollo BChain ([Hyperledger]). Viene poi descritto il protocollo *Redundant Bizantine Fault Tolerance*, introdotto in [Aublin et al., 2013] ed utilizzato in Hyperledger Indy, ed il protocollo *BFT SMaRT* ([Bessani et al., 2014]), sicuro ed efficiente per messaggi di piccola e media taglia per i quali la velocità di trasferimento è mostrata essere molto buona; infine viene descritto il *Ripple protocol consensus algorithm*, utilizzato nella piattaforma Ripple.

L'intera Sezione 1.2.4 è dedicata alla descrizione del *Proof of vote*, protocollo di tipo vote-based proposto in [Li et al., 2017], basato su un meccanismo di voto applicabile ad una Consortium blockchain. Tale protocollo è meno noto dei precedenti, ma sarà utilizzato in questa tesi nelle Sezioni 2.3 e 2.4. La Proof of Vote separa il diritto di voto dal diritto di produzione dei blocchi: il ruolo dell'esecuzione della produzione di un blocco è affidato ad un team affidabile e privo di leader, reclutato attraverso l'intera rete ed eletto a rotazione, mentre ogni blocco sarà verificato e votato da ogni compagnia del consorzio, in modo da allontanare eventuali membri disonesti. Inoltre, un unico blocco valido sarà generato dalla votazione, senza possibilità di fork ed il tempo di conferma delle transazioni è ottimizzato rispetto agli esistenti

algoritmi di consenso.

Nella Sottosezione 1.2.5 vengono esaminate forme ibride di algoritmo di consenso, primo fra tutti il protocollo introdotto in [King et al, 2012] ed utilizzato in PPCoin; per poter appendere il proprio blocco alla catena, un nodo deve effettuare una Proof of Work in modo da verificare una certa condizione che è funzione non solo della difficoltà come nella Proof of Work originale, ma anche del *coin age* di un output scelto (cioè il prodotto tra il valore dell'output ed il tempo di possesso): maggiore è la coin age e maggiore sarà il target, rendendo più semplice il lavoro di mining. Una debolezza di tale protocollo è data dal fatto che i nodi potrebbero essere tentati di non partecipare al sistema di verifica per accumulare più coin age. Per risolvere questo problema, il protocollo precedente è stato modificato in [Vasin, 2014] utilizzando il puro valore di stake al posto della coin age (ed adottato in Blackcoin), ed in [Ren, 2014] (adottato in reddcoin) abbinando la coin age con una funzione esponenziale di decadimento. Anche la Proof of Activity ([Benton et al., 2014]) utilizza sia la Proof of Stake che la Proof of Work, e nasce per migliorare il protocollo di Bitcoin, mentre il protocollo *fork-free hybrid consensus with flexible Proof of Activity*, proposto in [Liu et al, 2017], sfrutta tre tipi di algoritmi di consenso: Proof of Work, Proof of Stake e

Bizantine Fault Tolerance.

Nel descrivere i vari algoritmi di consenso sopra elencati, all'interno delle Sottosezioni sopra citate sono già stati evidenziati i loro punti di forza, le loro problematiche e la potenza avversaria tollerata nei vari casi; anzi, la trattazione, oltre ad offrire come già detto una panoramica sugli algoritmi di consenso esistenti, approfondisce via via i legami tra i vari protocolli, sottolineando come alcuni di essi nascano per migliorare i precedenti. Tuttavia nella Sezione 1.3 sono state messe in evidenza le principali caratteristiche e differenze tra i vari algoritmi di consenso, i vari tipi di blockchain adatti ad ognuno di essi, ed i confronti tra i protocolli esaminati per quanto riguarda performance, scalabilità e resilienza agli attacchi.

Dopo aver compreso la tecnologia blockchain e confrontato criticamente i vari algoritmi di consenso, si è passati ad analizzare l'utilità dell'impiego di tale tecnologia in simulazione.
Come osservato nella Sezione 2.1, in simulazioni volte ad ottimizzare un sistema sulla base di input noti e con parametri modificabili, vari simulatori potrebbero lavorare in parallelo e condividere tramite una rete blockchain

quella che per ognuno è la soluzione migliore al problema, unitamente ai valori dei parametri che la realizzano; una volta propagata una possibile soluzione di ottimizzazione, gli altri simulatori potrebbero verificarla utilizzando i relativi parametri e, se l'algoritmo di consenso utilizzato la eleggesse come migliore, essa verrebbe inserita in blockchain.

In simulazioni interoperabili distribuite, per esempio basate su *High Level Architecture*, l'utilizzo della tecnologia blockchain permetterebbe di confermare in modo formale le operazioni eseguite su un oggetto e quindi di evitare operazioni errate che possono rivelarsi dannose, alterando il risultato della simulazione. Per esempio, nel caso di una supply chain, e quindi di un sistema per sua natura distribuito e con molti parametri stocastici ed alto numero di interazioni, un approccio di Modeling and Simulation è un ben noto strumento di aiuto alle decisioni al fine di migliorare la gestione, permettendo di introdurre possibili effetti stocastici ([Bruzzone et al., 2005]); inoltre la sopra citata High Level Architecture permette di connettere diversi componenti del simulatore e di fornire una realistica rappresentazione del sistema, ancora più efficiente se combinata con l'introduzione di agenti intelligenti che meglio riprodurrebbero il comportamento dei differenti partecipanti alla supply chain: clienti, fornitori, trasportatori potrebbero essere rappresentati

da agenti intelligenti con corrispondente logica di operazione ed un all-seeing viewer potrebbe individuare eventuali criticità e suggerire come migliorare il sistema. Utilizzare una blockchain permetterebbe di salvare tutti i dati confermati relativi ad un oggetto, come il suo trasporto, cambio di proprietà, o una qualsiasi modifica.

Nella Sezione 2.2 viene proposta l'applicazione della tecnologia blockchain per la previsione della domanda di una supply chain, su due fronti: da un lato applicata nella *simulazione* di una supply chain, per garantire la certificazione del dato della simulazione e la fiducia reciproca tra le varie entità; dall'altro applicata ad una supply chain *reale*, per permettere l'identificazione e il tracciamento dei prodotti mediante informazioni convalidate e verificate, punto di partenza per la successiva creazione di un preciso modello di previsione della domanda e l'ottimizzazione della produzione e della logistica.
In particolare, viene dapprima suggerito l'utilizzo della tecnologia blockchain in un modello agent based di supply chain proposto in [Liang et al., 2006]. La scelta fatta è basata sulle seguenti considerazioni. Una supply chain comprende molti sistemi, legati alla produzione, al deposito, al trasporto e alla distribuzione, e allo scopo di una ottimizzazione globale è necessaria coordi-

nazione tra le varie entità: se l'intero sistema non è coordinato (cioè se ogni entità pensa alla propria ottimizzazione locale senza considerare l'impatto delle proprie decisioni sulle altre), ciò può portare ad una maggiore variazione tra la domanda e l'offerta a livello globale; per la coordinazione è necessaria la comunicazione tra le varie entità, e tale comunicazione deve essere approfondita e basata su informazioni *accessibili e trasparenti*, [Lee et al., 1997]. La mancanza di informazione tra i livelli di una supply chain è stata individuata da Forrester e da Sterman, rispettivamente in [Forrester, 1961, Sterman et al, 1992], come una delle cause dell'*effetto Bullwhip*, o effetto Forrester, definito da Forrester stesso come l'amplificazione della variabilità del segnale di domanda / ordine che si riscontra man mano che questo risale, da valle a monte, dal retailer al manufactorer, lungo la filiera logistica. In [Lee et al., 1997] gli autori aggiungono tra le cause dell'effetto Bullwhip gli errori nella previsione della domanda. Molti autori hanno studiato come migliorare l'accuratezza nella previsione della domanda, spesso evidenziando come ciò possa tamponare l'effetto Bullwhip. Molte ricerche però si sono focalizzate sulla previsione della domanda di un singolo livello della catena; in [Kimbrough et al, 2002, McBurney et al., 2002, Chen et al., 2000] gli autori hanno considerato i vari livelli della catena per la previsione della domanda,

ma hanno assunto che ogni livello adottasse lo stesso sistema di gestione dell'inventario, cosa che nella realtà succede di rado. In [Liang et al., 2006] gli autori utilizzano il *Real-coded Genetic Algorithm* (RGA) [Michalewicz, 1996] in quanto esso non richiede grandi quantità di dati e poiché la sua funzione obiettivo è il costo minimo totale: nel codice genetico ogni bit rappresenta un ordine per ognuna entità, quindi ogni ordine di ciascuna entità è considerato durante il processo di evoluzione, e il costo totale della supply chain è ottimizzato globalmente. Inoltre il modello di supply chain permette di usare nei vari livelli diversi sistemi di gestione dell'inventario.

Il modello agent based permette agli agenti di comunicare e di condividere con le altre entità le informazioni sulla domanda e l'assunzione fatta dagli autori è che ogni entità sia disposta a condividere in modo *onesto* le informazioni con le altre; sotto tale assunzione il costo totale è minimizzato. In tale contesto è evidente come l'applicazione della tecnologia blockchain potrebbe rendere fondata tale assunzione, garantendo la certificazione del dato della simulazione e la fiducia reciproca tra i control agent.

Per quanto riguarda invece l'applicazione della tecnologia blockchain ad una supply chain reale, anche se molte compagnie hanno già annunciato l'intenzione di utilizzare le blockchain per il tracciamento dei propri prodotti, tra cui

IBM ([Galvin, 2017]), De Beers ([Butcher, 2018]) e le compagnie Provenance e Agridigital [Bermingham, 2017], come osservato in [Bruzzone et al., 2018] i report riguardanti tale applicazione sono rari e generici e non sono ancora stati resi pubblici i relativi dettagli tecnici. Viene quindi da noi analizzato come poter applicare la tecnologia blockchain ad una supply chain. Si descrivono gli asset e le transazioni in questo contesto, i ruoli dei vari nodi partecipanti alla rete blockchain, e si evidenzia come ciò permetterebbe di: identificare un prodotto in modo univoco con informazioni convalidate e verificate; tracciare il prodotto consentendo a fornitori, produttori e distributori di fornire tutte le informazioni relative al tracciamento dello stesso nel ledger distribuito con automatica verifica di tali informazioni; segnalare prodotti sospettati di contraffazione o pericolosi; rilevare eventuali non osservanze di particolari regole di trasporto; segnalare eventuali difformità da regole contrattuali condivise dalla rete di business. Inoltre, l'applicazione della tecnologia blockchain ad una supply chain permetterebbe di avere dati sicuri riguardanti consegne e ordini, per cui consentirebbe di creare un preciso modello di previsione della domanda e dell'offerta, contribuendo quindi di ottimizzare produzione e logistica.

Analogamente a quanto visto per le supply chain, la tecnologia blockchain può essere un utile strumento per la *prevenzione* e *gestione* dei disastri naturali: da un lato si potrebbero certificare sulla blockchain tutti i dati relativi alle opere di pianificazione della prevenzione o riguardanti la gestione delle emergenze; dall'altro lato, le blockchain utilizzate in simulazione permetterebbero la certificazione del dato della simulazione stessa e garantirebbero la fiducia reciproca tra le varie entità rappresentanti il comune, la provincia, i Vigili del Fuoco, ecc. La prevenzione delle catastrofi naturali può essere condotta su due fronti: il primo riguardante investimenti con la costruzione di strade, argini di contenimento di fiumi, eccetera; il secondo riguardante una migliore regolazione della cementificazione in zone soggette a rischi idrogeologici, nivologici, sismici eccetera. Per quanto riguarda il primo punto, l'utilizzo della blockchain permetterebbe ai vari enti di certificare nel database distribuito le azioni fatte per pianificare la prevenzione dei disastri naturali e, in caso di eventi quali alluvioni, terremoti, eccetera, certificare come tali misure si siano rivelate utili o insufficienti: avendo uno strumento di tracciamento di tutte le azioni preventive e della misura in cui si siano rivelate utili o insufficienti in caso di realizzazione dell'evento temuto, si potrebbe creare un più preciso modello di prevenzione e utilizzarlo per ottimizzare la

pianificazione della prevenzione stessa.

Nella sezione 2.3 viene proposta l'applicazione della tecnologia blockchain per regolare la cementificazione in zone soggette a rischi naturali. La motivazione nasce purtroppo da eventi disastrosi che avrebbero risparmiato vittime qualora si fosse evitata la costruzione di fabbricati in zone a rischio. Un drammatico esempio è rappresentato dalla tragedia dell'Hotel Rigopiano, situato ai piedi del Gran Sasso e spazzato via da una slavina provocando 29 vittime: l'albergo non avrebbe dovuto essere edificato in quel luogo in quanto la zona era a rischio valanghe; inoltre è stato costruito sui detriti di valanghe precedenti. Purtroppo, anche in presenza di controindicazioni oggettivamente dimostrabili, spesso per motivi economici la cementificazione non viene arrestata. Da qui è nata l'idea di utilizzare la tecnologia blockchain per la gestione della cementificazione: la partecipazione di organizzazioni decentralizzate al processo decisionale riguardante il permesso o meno di costruire ed un sistema efficace di verifica possono evitare che interessi economici abbiamo la meglio su rischi oggettivi. Per realizzare una blockchain a tale scopo si è scelto l'algoritmo Proof of Vote (Sezione 1.2.4) che, ricordiamo, possiede un meccanismo tale da allontanare eventuali membri disonesti. La Sezione 2.3 quindi si conclude con la descrizione particolareggiata dell'utilizzo di una

consortium blockchain con protocollo di tipo Proof of Vote con lo scopo di autorizzare o meno le richieste di costruzione, che qui di seguito sintetizziamo: le organizzazioni del consorzio sono i comuni italiani accomunati dallo stesso rischio, per esempio nivologico; una transazione è una richiesta di costruzione, il cui mittente è il costruttore ed il destinatario il perito comunale, che la firma e la propaga nella rete blockchain se la ritiene fattibile e se la delibera comunale ha espresso parere favorevole; dopodiché esperti nominati a livello nazionale sono addetti alla creazione dei blocchi contenenti transazioni che ritengono fattibili; gli esperti comunali invece voteranno sia il blocco sia l'operato dell'esperto nazionale. Nel caso dell'hotel Rigopiano, ad esempio, la richiesta di costruire sarebbe dovuta dunque essere accettata, oltre a livello locale, anche dal nivologo esperto incaricato e da più della metà dei nivologi dei comuni italiani, quindi probabilmente la costruzione non sarebbe stata concessa.

La descrizione della consortium blockchain con protocollo Proof of Vote per la regolazione della cementificazione in aree soggette a rischi naturali ci ha suggerito un'altra idea di applicazione che, anche se esula dai fini di questa tesi, ci è sembrata degna di nota e quindi è stata comunque inserita nella Se-

zione 2.4. Come noto, per una banca la valutazione del valore di un immobile oggetto di richiesta di mutuo da parte di un perito è fondamentale: qualora il mutuatario non rispettasse le obbligazioni, se il mutuo erogato fosse superiore al valore dell'immobile, ciò comporterebbe alla banca una forte perdita. Viene quindi proposta l'applicazione di una consortium blockchain con protocollo Proof of Vote per il reclutamento di un perito per la valutazione di un immobile oggetto di richiesta di mutuo. In questo contesto, le organizzazioni del consorzio sono le filiali di una stessa banca; una transazione consiste nella valutazione di un immobile da parte del perito locale; nel database essa sarà corredata dalle informazioni che hanno portato a tale stima e sarà firmata dalla relativa filiale se ritenuta congrua. Tutte le transazioni saranno successivamente divulgate nella rete e ogni perito nazionale inserirà quelle ritenute valide nella propria transaction pool. Tramite l'algoritmo di Proof of Vote un perito nazionale sarà scelto in maniera casuale per confezionare il blocco successivo che, una volta creato, sarà votato dalle varie filiali, che voteranno anche il perito.

Capitolo 1

Blockchain e algoritmi di consenso

1.1 Breve descrizione delle blockchain

Verso la fine degli anni '90 sono nati i sistemi *peer-to-peer*, cioè modelli di architettura logica di rete in cui ogni nodo è equivalente (peer), in capacità e responsabilità, a tutti gli altri, in contrapposizione agli esistenti modelli centralizzati che concentrano il potere in un unico nodo centrale (Figure 1.1, 1.2, 1.3).

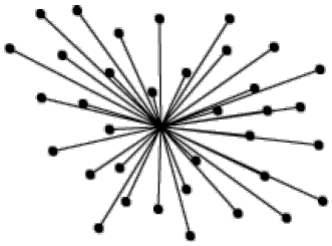

Figura 1.1: Schema di una rete centralizzata

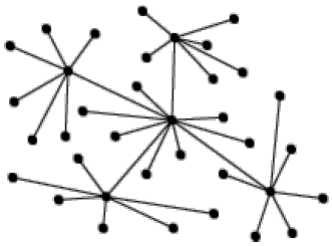

Figura 1.2: Schema di una rete decentralizzata

Una blockchain (Figura 1.4) è un *distributed ledger* ossia un database distribuito che, sfruttando la tecnologia peer-to-peer, gestisce un elenco di dati sempre crescenti mediante una procedura di codifica a blocchi (elemento caratterizzante della tecnologia, da cui il nome) ideata per garantire a tali dati condivisi pubblicamente sicurezza e immutabilità mediante l'uso di strumenti crittografici e di un sistema efficace di verifica.

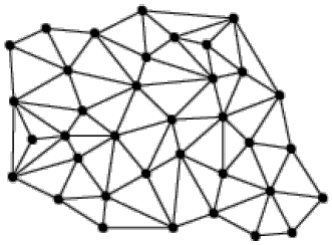

Figura 1.3: Schema di una rete distribuita

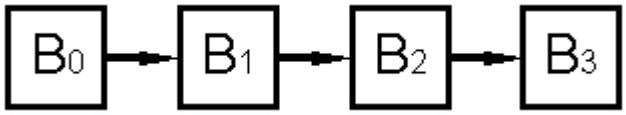

Figura 1.4: Schema di una blockchain

La struttura di ogni blocco della catena (ad eccezione del primo blocco, detto *genesis block*) che compone il distribueted ledger consta generalmente di 4 campi (figura 1.5): la sua dimensione in byte (*Block size*), l'intestazione (*Block Header*), il numero di transazioni che contiene (*Transaction Counter*), e le transazioni vere e proprie, che corrispondono ad operazioni tra oggetti della blockchain. Inoltre ogni blocco è marcato temporalmente dal *timestamp*, inserito nell'Header Block per fornire in maniera certa il momento della sua creazione, ed è identificato da un hash (*block hash*) generato da un algoritmo crittografico, e poiché nell'intestazione del blocco stesso

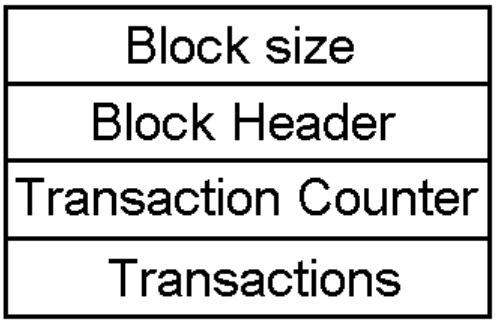

Figura 1.5: Struttura di un blocco

c'è un puntatore (*previous block hash*) all' hash dell'unico blocco che lo precede nella catena (*parent block*), un' eventuale modifica del parent block comporterebbe una modifica del blocco attuale e di tutti i blocchi ad esso successivi; ciò assicura che tanto più numerosi sono i blocchi che seguono un dato blocco nella catena, tanto più il blocco stesso può considerarsi immutabile [Antonopoulos, 2017].

Vogliamo ora capire come un blocco della blockchain viene aggiunto alla catena; per farlo dobbiamo ripercorrere la storia delle transazioni che esso contiene.

Ogni transazione presente nel blocco è stata dapprima creata da un uten-

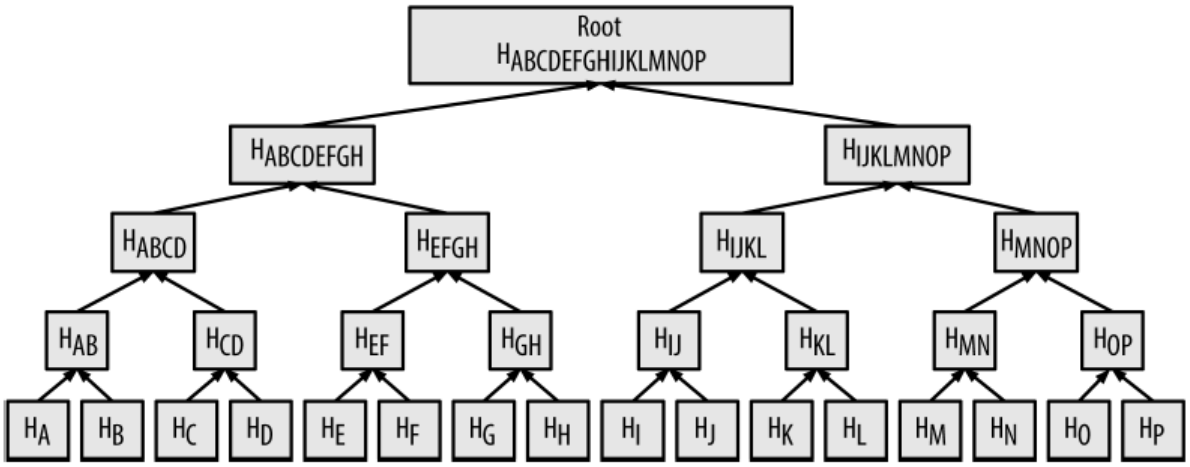

Figura 1.6: Merkel tree

te, identificato pubblicamente da un indirizzo generato dalla propria chiave pubblica, e successivamente è stata firmata digitalmente, assegnando in input ad una *sign function* (per esempio in Bitcoin il protocollo ECDSA, Ellyptic Curve Digital Signature Algorithm [Johnson et al., 2001]) la propria chiave privata. La transazione è stata poi trasmessa alla rete senza bisogno di essere cifrata in quanto l'utilizzo sopra descritto di chiavi pubbliche e private garantisce l'assenza di informazioni private (anche se è stato dimostrato che la blockchain non può garantire totale riservatezza né sulle transazioni [Meiklejohn et al., 2013, Kosba et al., 2016] né sugli utenti [Barcelo, 2014, Biryukov et al., 2014]); una volta trasmessa, ogni nodo che l' ha ricevuta ne ha controllato la *validità* seguendo un'opportuna lista di

criteri e, appurata la sua validità, l'ha propagata nella rete (*flooding*) ai nodi ad esso connessi. Ogni nodo che ha ricevuto la transazione valida, dopo averne verificato indipendentemente la validità, ha costruito un *memory pool* di transazioni valide ma non confermate. Speciali nodi hanno aggregato la nostra transazione, insieme alle altre valide che hanno ricevuto, in un blocco candidato a diventare il nuovo blocco da appendere alla catena (*candidate block*); a questo punto, poiché ognuno di questi nodi speciali ha proposto ai nodi ad esso connessi il proprio blocco candidato, si è reso necessario un accordo (*consensus*) tra tutti i nodi su quale blocco aggiungere tra i blocchi proposti, visto che uno solo di essi può essere aggiunto alla catena; è stato quindi applicato il *consensus algorithm* (sezione 1.2) adottato dalla blockchain che ha eletto il nostro blocco, che quindi ora fa parte della blockchain.

Come abbiamo intuito, concetti chiave della tecnologia blockchain che consentono di garantire la sicurezza dei dati sono il consenso condiviso e la crittografia, per una efficace verifica delle transazioni contenute nei blocchi della catena.

Gli algoritmi di consenso condiviso risolvono il problema della sincronizzazione nei database distribuiti e saranno analizzati nella Sezione 1.2.

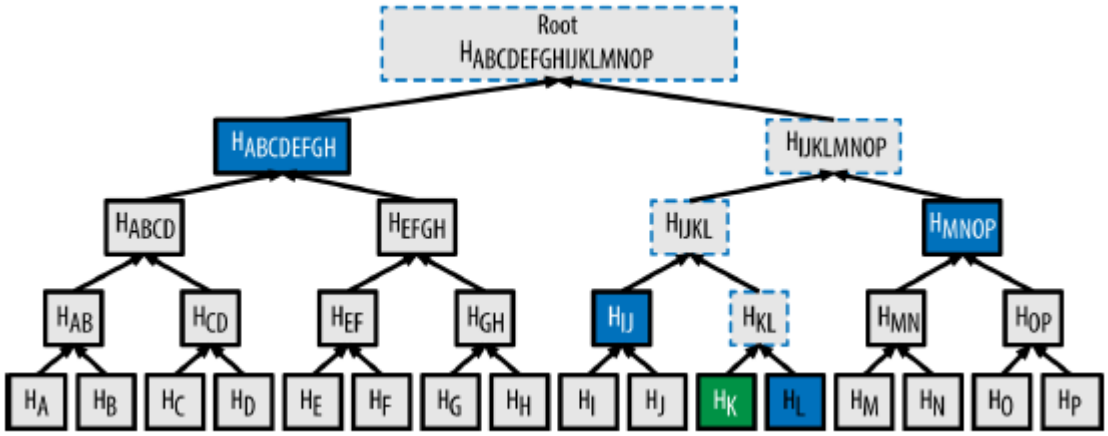

Figura 1.7: Merkel path (in blu) rispetto al nodo H_K

Quanto agli strumenti crittografici, abbiamo già incontrato l'uso di sign function per ottenere firme digitali e accennato all'uso di hash function per la creazione del block hash, che permette di collegare ogni blocco al proprio parent block; ma le funzioni hash sono utili nella blockchain anche per indicizzare efficientemente grandi quantità di dati e permettere anche ai nodi che non hanno la copia dell'intera blockchain di riuscire comunque a verificare se una determinata transazione è presente in un blocco e di verificarne l'integrità. Ciò è possibile grazie all'utilizzo degli alberi di Merkel [Merkel, 1979], strutture dati che consistono in alberi binari bilanciati costruiti a partire dalle foglie dal basso verso l'alto ricorsivamente, concatenando di volta in volta

gli hash dei nodi figli (Figura 1.6, da [Antonopoulos, 2017]): nel caso delle blockchain, le foglie dell'albero di Merkel di un blocco sono le transazioni del blocco, inserite nell'albero tramite i loro hash quali nodi del livello più basso dell'albero (nel caso di numero di transazioni dispari l'albero è bilanciato duplicando una transazione); i nodi di ogni livello superiore dell'albero sono ottenuti hashando la concatenazione degli hash dei due nodi figli situati nel livello precedente, fino ad arrivare ad un unico hash detto *Merkel root*, memorizzato nell'header block del blocco stesso.
Dato un albero di Merkel, per dimostrare che una foglia è inclusa nell'albero, nota la radice, è sufficiente utilizzare un percorso di autenticazione (*Merkel path*) che colleghi l'hash di tale foglia alla radice, calcolando quindi solo un numero limitato di hash (Figura 1.7, da [Antonopoulos, 2017]); in una blockchain, dunque, ogni nodo può verificare se una certa transazione K appartiene ad un blocco scaricandone solo il block header e recuperando il Merkel path da un nodo che ha il database completo della blockchain.

La tecnologia blockchain è nota soprattutto per la sua utilizzazione in Bitcoin [Nakamoto, 2008], ma da allora è stata utilizzata in molte applicazioni; esistendo quindi diverse tipologie, possiamo classificare le blockchain

in base ad opportuni criteri.

Swan ha definito tre categorie di blockchain in base al loro utilizzo [Swan, 2015]:

- blockchain 1.0;
- blockchain 2.0;
- blockchain 3.0.

Con blockchain 1.0 si indicano le applicazioni delle blockchain alla valuta e alle criptovalute, e quindi legate al contante (come il trasferimento di denaro) ed ai sistemi di pagamento digitale; con blockchain 2.0 si indicano le blockchain utilizzate nei contratti, e quindi aventi applicazioni in ambito economico, finanziario e di mercato, ma più estese di una semplice transazione di denaro (e quindi azioni, obbligazioni, finanziamenti, smart property e smart contract); infine con blockchain 3.0 ci si riferisce alle applicazioni delle blockchain ad ambiti al di fuori dell'ambito finanziario (per esempio il monitoraggio e il controllo di provenienza di un oggetto).

Un altro criterio di distinzione comunemente utilizzato è quello relativo al permesso (necessario o non) di partecipare alla rete:

- blockchain *permissionless* (o *unpermissioned*);

- blockchain *permissioned*.

Mentre le blockchain di tipo permissionless (o unpermissioned) sono aperte a chiunque voglia farne parte (e per questo sono anche dette *public blockchain*), sia per disporre dei dati del registro pubblico che per contribuire all'aggiornamento dello stesso, senza previa autorizzazione, le blockchain di tipo permissioned richiedono un'autorizzazione per poter partecipare alla rete. In queste ultime dunque sono definite speciali regole per l'accesso e la visibilità dei dati, e solo alcuni nodi (detti *trusted*) hanno il compito di validare le transazioni ed i blocchi, a seconda del consensus algorithm utilizzato; inoltre, a seconda se i nodi trusted appartengono ad una sola organizzazione o a diversi consorzi, possono essere a loro volta suddivise rispettivamente in *private Blockchain* e *consortium Blockchain*[Buterin, 2015].
Charamente mentre nelle blockchain pubbliche la decentralizzazione è totale, nelle consortium blockchain si ha una parziale centralizzazione, che invece diventa totale in quelle private; di qui si evince che la probabilità di manomissione dei dati, bassa nelle blockchain pubbliche, aumenta negli altri due tipi, proprio per la presenza di una o più organizzazioni dominanti.
Per contro, l'efficienza delle blockchain pubbliche è minore, in quanto la velocità nel processo di propagazione e validazione delle transazioni è inver-

samente proporzionale al numero dei nodi validatori.

Le blockchain permissioned possono essere utilizzate da istituzioni e grandi imprese, le quali devono gestire lunghe filiere o catene di approvvigionamento con fornitori e subfornitori, banche, enti sanitari, Pubblica Amministrazione. In generale le blockchain permissioned risolvono il problema del rispetto di norme e regole: gli algoritmi di consenso applicati nelle blockchain pubbliche si fondano su informazioni pubbliche incontrollabili; in una blockchain di tipo permissioned si può limitare la facoltà di verifica della transazioni e dei blocchi pur mantenendo una decentralizzazione nel processo decisionale e nel rispetto di regole.

Esempi di blockchain permissioned sono Hyperledger Fabric [Hyperledger] e Tendermint [Tendermint].

Esempi di blockchain permissionless sono Bitcoin, Ethereum e Nxtcoin.

Vedremo quali tipi di consensus algorithmsono preferibili a seconda che la blockchain sia permissioned o permissionless nella prossima sezione (Sezione 1.2).

1.2 Algoritmi di consenso nelle blockchain: tipologie, affidabilità e resilienza

Come già anticipato, i protocolli di consenso sono lo strumento chiave attraverso il quale i nodi si accordano su quale blocco aggiungere alla catena e che quindi garantiscono l'uniformità della blockchain, risolvendo il problema della sincronizzazione nei database distribuiti.

A seconda di come tale accordo viene raggiunto, essi possono essere suddivisi in due grandi categorie: nel primo gruppo il blocco sarà aggiunto alla catena dal nodo che avrà *dimostrato* in qualche modo (differente nei vari algoritmi) di essere il più qualificato degli altri nel farlo (*proof-based algorithm*); nel secondo gruppo un blocco sarà aggiunto solo dopo che i nodi (o alcuni di essi, a seconda dello specifico protocollo) si saranno scambiati opinioni riguardanti i propri risultati di verifica indipendente (*vote-based algorithm*).

I proof-based consensus algorithm saranno preferibili nelle blockchain pubbliche, mentre nelle blockchain di tipo permissioned saranno preferibili i vote- based essendo minore il numero di nodi trusted.

I principali esponenti di proof-based algorithm sono il *proof of work* (PoW) ed il *proof of stake* (PoS), mentre il principale esponente di vote-based algorithm è il *Pratical Byzantine Fault Tolerance* ($PBFT$): a partire da questi tre tipi di protocollo molti studiosi hanno proposto alcune varianti, ma mantenendo i concetti chiave proposti da essi.

1.2.1 Proof of Work e sue varianti

La Proof of Work (PoW) è un algoritmo nato per dimostrare di aver impiegato delle risorse risolvendo un complesso problema computazionale, caratterizzato dalla asimmetria (deve essere difficile trovare la soluzione del problema ma facile verificarne la soluzione), dalla possibilità di modificare la difficoltà del problema, e dall'impossibilità di prevedere la soluzione.

Ideato da Adam Back [Back, 1997], è utilizzato in Bitcoin per il processo di *mining*: per avere il diritto di aggiungere un blocco alla catena, un nodo (*miner*) deve essere il primo a risolvere un difficile problema matematico basato su un algoritmo crittografico di hashing.

Più precisamente il miner deve trovare un input per la funzione hash *double-SHA256* [SHA256, 2016] in modo che l'output risultante abbia un numero di zeri iniziali fissato dal *difficulty target*, che in Bitcoin è aggiornato ogni

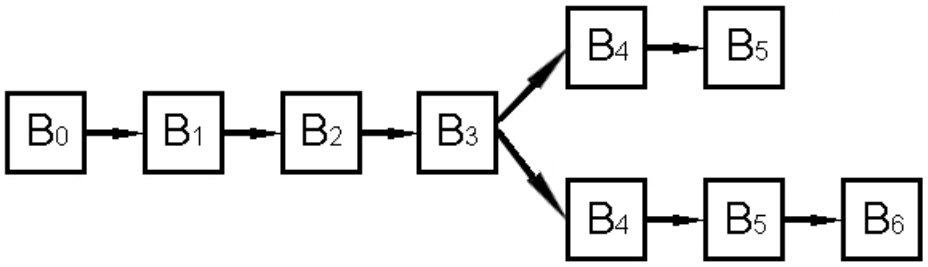

Figura 1.8: Fork in Blockchain

2016 blocchi aggiunti: per trovare la soluzione il miner prova a concatenare l'header del blocco con un parametro (*nonce*) fino a trovare il nonce giusto che permettere di avere l'output richiesto.

Trovare il nonce che funzioni (cioè effettuare il mining sul blocco) è molto laborioso (il numero dei tentativi richiesti è nell'ordine dei quadrilioni, in Bitcoin, con la difficoltà attuale [Antonopoulos, 2017]): per questo si parla di proof of work.

Il miner che ha vinto la gara di mining propaga il blocco ai nodi peers, i quali verificano la validità del blocco stesso prima di trasmetterlo a loro volta nella rete, effettuando dunque una validazione indipendente, concetto chiave del consenso condiviso.

Tuttavia può succedere che il mining sia effettuato quasi simultaneamen-

te da due diversi miner che quindi, l'uno ignaro dell'altro, trasmettono il proprio blocco agli altri nodi; questo porta ad una biforcazione della catena (*fork*, Figura 1.8). In tal caso i miner continueranno a minare blocchi sui fork fino a quando uno dei due rami diventerà più lungo; a quel punto tutti i nodi riconosceranno quel ramo come il principale della blockchain ed il pieno consenso è ottenuto; per le transazioni contenute nel blocco situato nell'altro ramo del fork ripartirà l'iter descritto nella sezione precedente.

Un primo lato negativo della PoW è il consumo energetico, vista la grande quantità di calcoli che i miner devono effettuare per minare un blocco: attualmente l'energia consumata da Bitcoin per il processo di PoW da Bitcoin supera quella dell'Irlanda ([Hern, 2018]), e in [motherboard] si stima che nel 2020 equivarrà a quella della Danimarca intera.

Oltre alla non trascurabile problematica energetica, l'alto costo di hardware ed elettricità necessari ai miner per poter competere alla gara di mining, spesso porta alla creazione di mining pool [CoinDesk Inc, 2014], nelle quali i miner condividono la potenza di hashing; in tale scenario, se una mining pool arriva a controllare la maggioranza della potenza di hashing, può verificarsi un attacco al consenso (*51% attack*).

In un attacco al consenso viene deliberatamente creato un fork che invalida blocchi validi e riconverte il ledger alla nuova catena, proprio in virtù della maggior potenza di mining. Un tale attacco può essere compiuto mediante la cosiddetta *selfish mining strategy*: un miner nasconde alcuni blocchi minati e li propaga in rete solo successivamente in modo tale da creare un fork in cui il nuovo ramo è più lungo. Nel caso di blockchain utilizzate in campo monetario un attacco al consenso può essere di tipo *double spending attack* [double spending, 2017]; in generale viene attuato per rendere invalide transazioni già validate.

Chiaramente, come osservato in [Antonopoulos, 2017], un attacco al consenso può solo colpire i più recenti blocchi visto che, come detto nella sezione precedente, realizzare un fork ad alta profondità richiederebbe una potenza di calcolo troppo grande.

È stato provato [Eyal et al., 2014] che per eseguire un attacco al consenso in una blockchain che utilizza la PoW è sufficiente possedere il 25% della potenza di hashing.

Un altro problema deriva dalla latenza delle transazioni, cosa che rende la PoW inadatta a seconda del tipo di transazioni della blockchain (per esempio per il pagamento real time, [Eyal et al., 2016]).

Alcune varianti della PoW sono state proposte, ciascuna per risolvere uno dei problemi sopra elencati.

In [Eyal et al., 2014, Miller et al., 2015] troviamo proposte per prevenire la formazione di mining pool, mentre in [Tromp, 2014] troviamo una proposta per ovviare al problema dei costi elevati di elettricità e di hardware, che consiste nel sostituire la ricerca del nonce giusto con quella di un opportuno grafo costruito utilizzando la funzione di hash Cackoo, in quanto tale ricerca richiede meno sforzi computazionali rispetto all'originale.

La variante della PoW utilizzata da Primecoin, invece, richiede di certificare il lavoro svolto senza utilizzare nessuna funzione di hash, ma cercando piuttosto una *catena di Cunningham* utilizzando un metodo ideato da King [King, 2013]; l'origine (media tra i primi due valori) della catena deve essere divisibile per l'hash del blocco che si vuole verificare (questa richiesta serve per non riutilizzare i risultati ottenuti in precedenza), e la lunghezza della catena deve essere superiore ad un lower bound fissato. Gli altri nodi, per verificare la primalità utilizzeranno i test di Eulero-Lagrange-Lifchitz [Lifchitz, 1998] e quello di Fermat [Caldwell, 2002] (basato sul piccolo teorema di Fermat).

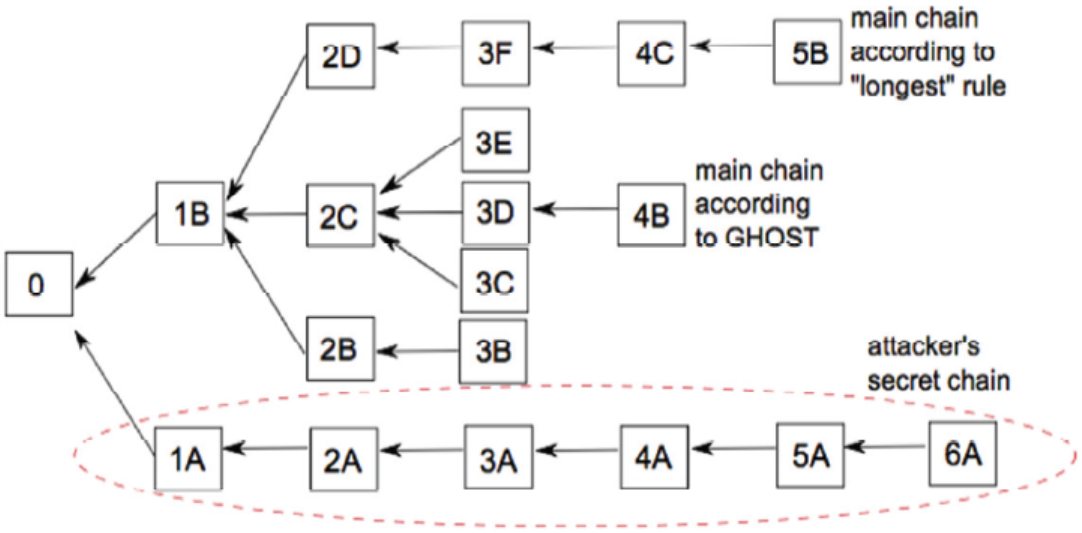

Figura 1.9: Strategia Ghost in presenza di fork

Per il problema della latenza, in [Eyal et al., 2016] viene proposto l'utilizzo di microblocchi per permettere una verifica più veloce delle transazioni. Ovviamente, più veloce è la generazione dei blocchi, più la sicurezza può essere compromessa.

La soluzione Greedy Heaviest-Observed Sub-Tree (*GHOST*) [Ghost, 2015] è nata per cercare di trovare un compromesso tra velocità e sicurezza: tale strategia propone di mantenere tutti i fork e di scegliere la catena derivante non dal ramo più lungo ma da quello con maggior contributo di PoW (Figura

1.9, da [Ghost, 2015]). In tal modo tale ramo sarebbe scelto anche tra un eventuale ramo derivante da un attacco al consenso (che sarebbe più lungo ma con meno PoW).

In [Shoker, 2017] l'autore ha proposto la *Proof of eXercise* (PoX), un'alternativa sostenibile (come da esso stesso descritta) alla PoW originale, in cui un *eXercise* è un real world matrix-based scientific computation problem. Lo spirito che ha mosso l'autore è stato quello di creare una variante dell'algoritmo PoW con lo scopo di ottenere output utili per evitare sprechi di risorse: tra le applicazioni al mondo reale si evidenziano problemi riguardanti l' image processing e il confronto e sequenziamento di DNA ed RNA.

Anche lo schema proposto in [Ball et al., 2017] nasce come alternativa alla PoW con lo scopo di rendere il lavoro svolto utile per risolvere problemi pratici, unendo dunque l'utilità alla difficoltà, già presente nella PoW originale. In particolare gli autori costruiscono lo schema *Proof of Useful Work* (uPoW) basato sul problema OV dei vettori ortogonali ([Gao et al., 2016]): il calcolo investito in questo schema può essere utilizzato per contribuire a risolvere tale problema.

Gli autori sottolineano inoltre come lo schema possa essere adattato anche ad altri problemi reali quali, tra gli altri, i problemi 3SUM (che si chiede, dati A, B e C insiemi di cardinalità al più n, se esiste una terna $(a, b, c,) \in A \times B \times C$ tale che $a + b = c$, [Baran et al., 2008]) e All-Pairs Shortest Path (che cerca il cammino più corto che lega ogni coppia di vertici di un grafo con n vertici, connesso non pesato e non orientato, [Seidel, 1992]).

Hyperledger Sawtooth, come Hyperledger Fabric, permette all'utente di scegliere l'algoritmo di consenso utilizzato e tra questi propone l'algoritmo *Proof of Elapsed Time* (PoET) [Rilee, 2018], secondo il quale ogni partecipante alla rete blockchain attende una certa quantità casuale di tempo; il primo partecipante a terminare l'attesa creerà il nuovo blocco. Per fare in modo che i partecipanti scelgano effettivamente un tempo di attesa casuale e che il vincitore abbia effettivamente finito di aspettare il tempo dovuto, viene utilizzato un pacchetto di istruzioni chiamato Intel Software Guard Extensions (SGX).

Partendo dall'osservazione che la PoW può essere vista come un modo di dimostrare di avere investito un notevole sforzo di risorse in relazione ad

una certa affermazione, in [Dziembowski et al., 2013] gli autori propongono l'algoritmo di *Proof of Space* (PoSpace) in cui lo sforzo computazionale della PoW originale viene sostituito dalla risorsa costituita dallo spazio del disco. La proposta degli autori è parzialmente motivata dal fatto che spesso gli utenti hanno a disposizione un significativo spazio libero nel disco, mentre non tutti hanno la potenza computazionale e la possibilità di dispendio energetico richiesti dalla PoW originale.

Una PoSpace è un protocollo tra un *prover* P e un *verifier* V (gli autori illustrano come esempio il caso in cui V sia un'organizzazione che offre un servizio mail libero) che consta di due fasi: una prima fase di inizializzazione dopo la quale P avrà immagazzinato una certa quantità di dati mentre V detiene solo alcune informazioni; dopo un certo tempo V potrà inizializzare la seconda fase, detta *proof execution*, alla fine della quale V delibererà rifiuto o accettazione. Nell'esempio di cui sopra, per evitare che qualche utente P registri un alto numero di indirizzi fake per spamming, V potrebbe richiedere agli utenti di dedicare una quantità di spazio nel disco non irrisoria, per esempio 100GB, per ogni indirizzo registrato ed occasionalmente V potrebbe applicare la PoSpace per verificare che l'utente P abbia realmente dedicato lo spazio del disco come dovuto. Per rendere piccola la complessità della

comunicazione faranno generare localmente un file a P durante la fase di inizializzazione e la fase di esecuzione della PoSpace sarà non dispendiosa né per P né per V. Tecnica chiave nella costruzione degli autori è quella del *pebbling*, utilizzata tipicamente in pebbling game giocati su un grafo orientato. In realtà, come osservato in [Ateniese et al., 2013], in una prima versione di Proof of Space presentata dagli autori di [Dziembowski et al., 2013] ad un Workshop tenutosi a Varsavia, non erano utilizzate tecniche di pebbling, utilizzate per prime in [Ateniese et al., 2013].
In [Park et al., 2015] viene descritto un protocollo per criptovalute basate sulla PoSpace.
La PoSpace è anche nota come *proof of capacity* [Andrew, 2018] ed è implementata in Burstcoin [Andrew, 2018].

In [Bowers et al., 2009] gli autori propogono un quandro teorico per la progettazione di una *Proof of Retrievability* (PoR), cioè una prova compatta da parte di un prover P ad un verifier V che un certo file di destinazione è intatto, nel senso che V può recuperarlo completamente; un PoR è utile per sistemi di archiviazione remota ad alta affidabilità in quanto ha anch'esso una complessità di comunicazione inferiore rispetto alla trasmissione del

file stesso. La costruzione proposta migliora le precedenti PoR proposte in [Juels et al., 2007] e [Shacham et al., 2008].

La *Proof of Burn* (PoB) nasce per ovviare al problema del dispendio energetico necessario per il mining con la PoW originale: per avere più potenza di mining si devono bruciare delle monete (*burning coin*), cioè spedirle ad un *burn adress*, ovvero uno specifico indirizzo di cui nessuno è proprietario e tale che tutte le monete a lui spedite sono perse per sempre (da cui il nome bruciate). Ciò avviene tramite delle *burn transaction*, speciali transazioni indirizzate al burn adress, che contengono burnt coin. Come asserito dallo stesso ideatore della PoB, Iain Stewart ([Stewart, 2012]), le monete bruciate sono vere e proprie piattaforme di mining: bruciare monete è come acquistare un impianto di estrazione virtuale la cui grandezza è proporzionale alle monete bruciate e che dà il potere di minare blocchi.
In [P4Titan, 2014] viene descritta Slimcoin, una criptovaluta che inizialmente utilizza la PoW per la distribuzione iniziale di monete, poi affianca PoB e PoS per dare sicurezza alla rete peer to peer senza dispendio energetico.

In [Milutivonic et al., 2016] è descritto un algoritmo di consenso chiama-

to *Proof of Luck* (PoL).

In ogni round ogni partecipante riunisce le transazioni ricevute dagli altri in un nuovo blocco e lo aggiunge alla propria catena corrente. Ad ogni blocco creato viene assegnato un valore numerico casuale (lucky). Dopodiché ogni partecipante trasmette agli altri la propria nuova catena ed un algoritmo (Luck algorithm) calcola il valore numerico (luck) di ogni catena sommando i valori numerici di ogni blocco. La catena con luck maggiore sarà riconosciuta come la catena principale.

Il protocollo non richiede sincronizzazione: i round dei partecipanti non sono sincronizzati, ma è il protocollo stesso che sincronizzerà i round.

Gli autori mostrano che il protocollo assicura bassa latenza delle transazioni, potere decentralizzato di mining, e difficoltà di attacchi in quanto un attacker dovrebbe essere fortunato per riuscire ad effettuare i propri intenti.

1.2.2 Proof of Stake e sue varianti

Una prima proposta di Proof of Stake (PoS) risale al 2011 [bitcoin forum, 2011]. L'algoritmo di PoS nasce per ovviare al problema della PoW legato al dispendio energetico e alla necessità di possedere hardware specializzati: come

già detto, nella PoW, ammesso che tutti i nodi seguano il protocollo, la probabilità di minare un blocco è proporzionale alla potenza computazionale (anche se tale proporzionalità cambia nel caso di deviazioni dal protocollo, per esempio nel caso di selfish mining strategy, [Sapirsthein et al , 2015, Eyal et al., 2014]); con la PoS la probabilità di aggiungere un blocco alla blockchain è proporzionale all'interesse (*stake*) e non più alla potenza di estrazione: ad esempio, in Nextcoin se un *forger*(l'analogo del miner) possiede una percentuale p delle monete totali in circolazione, avrà una probabilità di creare un nuovo blocco pari a p. Oltre al risparmio energetico, l'idea di base è quella per cui chi possiede più ricchezza dovrebbe essere più fidato; quindi dovrebbe essere meno probabile un suo attacco alla rete.

Chiaramente, se con la PoW i miner con meno potenza di calcolo sono svantaggiati e con poca probabilità di effettuare il mining su un blocco, nella PoS la disparità rimane, anche se ora legata all'interesse.

Notiamo fin da subito che la traduzione di *stake* in *interesse* non è casuale: nel caso di criptovalute sarà sinonimo di quota, ma in generale può avere significati diversi.

Ovviamente non solo la quantità di interesse può incidere sulla scelta del prossimo nodo (*leader*) che aggiungerà il blocco (altrimenti colui che possie-

de più stake sarebbe l'unico a costruire la catena), quindi in generale nella PoS è presente anche un altro fattore che dipende dalle varie versioni e che principalmente è stocastico.
I vari tipi di PoS proposti differiscono proprio nella randomizzazione che porta all'elezione del leader.

Il protocollo PoS utilizzato in Nextcoin genera ogni minuto in modo casuale un numero naturale positivo i minore o uguale al numero totale di Nxt (monete coniate nel blocco genesi); il nodo che possiede l'Nxt con indice i ha il diritto di propagare il proprio blocco nella rete per appenderlo alla catena.

Anche il protocollo puramente PoS denominato *Chains of Activity* (CoA) e proposto in [Benton et al., 2016] genera un indice i, minore stavolta del numero totale di satoshi in circolazione; ma la sua generazione è più complicata, in quanto è il risultato di una funzione di hash il cui input dipende, oltre che da un numero random, anche dal numero di blocchi esistenti nella catena e dai loro hash (con l'utilizzo di una certa funzione detta *comb*). Dopodiché il protocollo prevede di seguire la *follow-the-satoshi procedure*, già descritta in [Benton et al., 2014]: si cerca il blocco che ha coniato l'i-simo satoshi e, esa-

minando tutte le transazioni presenti nella blockchain che includono questo satoshi, si cerca il suo ultimo proprietario, che sarà colui che avrà la fortuna di poter appendere un nuovo blocco alla catena. Il protocollo prevede anche la formazione di una lista nera in cui inserire chi per tre volte consecutive, nononostante l'opportunità di creare un blocco, non sia riuscito a farlo. Notiamo qui che la *follow-the-satoshi procedure* potrebbe essere adattata anche in blockchain non legate a criptovalute.

Per rendere la randomizzazione dell' elezione il meno esposta possibile alla *grinding vulnerability*, ovvero alla manipolazione da parte di avversari che, simulando il protocollo, potrebbero predire il calcolo e quindi condizionare l'elezione, è stato proposto il protocollo *Ouroboros* [Kiayias et al., 2017], un sistema di PoS *provably secure*, come definito dagli stessi autori. In tale protocollo in ogni intervallo di tempo (epoch) una commissione di stakeholders attua il coin flipping che determina l'elezione del leader ma anche la commissione che attuerà il coin flipping nella fase successiva. In [Kiayias et al., 2017] gli autori sottolineano che anche se in [Benton et al., 2016] viene dimostrato che il protocollo CoA e più sicuro degli esistenti protocolli rispetto ad alcuni attacchi, Bentov ed i coautori non forniscono un modello formale per l'analisi

dei protocolli PoS e che le prove di sicurezza non si basano su precise definizioni; quindi, oltre a proporre Ouroboros, forniscono un modello generale di costruzione di protocollo PoS e forniscono argomentazioni formali che coinvolgono probabilità, combinatoria e teoria dei giochi per dimostrare la sicurezza del protocollo proposto. In particolare viene proposto un nuovo meccanismo di incentivazione alla partecipazione al protocollo che viene dimostrato essere un equilibrio di Nash, cosa che limita attacchi come il *block withholding* (selfish mining strategy) [Eyal et al., 2014, Sapirsthein et al , 2015].

Nel *delegated proof of stake* (dPoS) [Larimer, 2014] gli stakeholder eleggono una commissione di delegati che avranno il compito di verificare le transazioni, generare e validare i blocchi ed inserirli nella catena. Lo scopo della variazione è principalmente quello di accelerare i tempi di conferma delle transazioni e di validazione dei blocchi, in quanto i nodi addetti a tale compito sono minori. La potenza di voto è proporzionale allo stake. La commissione non è fissa ma viene cambiata continuamente; inoltre i membri della commissione in carica possono essere rimossi, sia in caso di non adempimento del compito, sia in caso di comportamento disonesto.

Nella sottosezione 1.3 verranno confrontati gli algoritmi di tipo PoW e di tipo PoS.

1.2.3 Bizantine Fault Tolerance e sue varianti

Già da decenni è stato affrontato il problema del consenso nei sistemi distribuiti dovuti alla presenza di *crashing nodes* o di *faulty nodes*, cioè di sistemi *Bizantine Fault Tolerant*.

Il nome deriva dal famoso problema dei generali bizantini ([Lamport et al., 1982]) che di seguito ricordiamo. Un'armata bizantina si suddivide in n gruppi, ciascuno guidato da un proprio generale, per attaccare il nemico da più lati, ma l'attacco deve avvenire contemporaneamente, altrimenti fallirebbe. Il problema è che se ci sono dei generali traditori, essi potrebbero far fallire l'accordo sul momento in cui procedere e di conseguenza anche l'attacco. In [Lamport et al., 1982] viene provato che per tollerare f generali traditori, sono necessari almeno $2f + 1$ generali onesti.

Tale risultato, applicato ad un contesto di consensus algorithm in una blockchain, può essere tradotto nella seguente *trust assumption* (che delinea l'ambito nel quale il protocollo può dare garanzie): dati n nodi indipendenti del sistema, se ci sono f nodi scorretti (*faulty*) ce ne devono essere almeno $2f+1$

che operano correttamente. Equivalentemente i nodi disonesti devono essere meno di $\frac{n}{3}$.

Nelle blockchain che utilizzano algoritmi di consenso di tipo Bizantine Fault Tolerance (BFT), le transazioni ed i blocchi devono essere verificati da tutti i nodi validatori insieme in modo da avere univocità nel consenso; tali nodi devono dunque essere noti per poter comunicare tra loro prima di decidere se aggiungere un blocco alla catena o meno. Per questo vengono per lo più adottati nelle blockchain di tipo permissioned.

Come sottolineato in [Cachin et al, 2017], per creare sistemi distribuiti resilienti realistici è preferibile adottare l'assunzione di *eventual synchrony*, introdotta da ([Dwork et al., 1988]): il modello di rete è asincrono, per cui la diffusione dei messaggi tra i nodi può essere ritardata arbitrariamente, ma entro un certo intervallo di tempo non noto diventa sincrona e tutti i messaggi sono distribuiti.

In [Duan et al., 2014] gli autori suddividono i protocolli BFT in due categorie: *broadcast-based* e *chain-based*.

La differenza sta nelle performance, in quanto i protocolli appartenenti alla seconda categoria mirano a raggiungere flussi più elevati a spese di una

maggiore latenza; tuttavia al crescere del numero dei nodi i protocolli di tipo chain-based possono ottenere latenza minore rispetto a quelli di tipo broadcast-based. Per contro i protocolli di tipo chain-based sono meno resilienti ai guasti.

Il principale esponente di protocollo broadcast based è il *Pratical Byzantine Fault Tolerance* PBFT), introdotto in [Castro et al., 2002]. In tale protocollo i nodi inviano le transazioni agli altri nodi che le valideranno e le trasmetteranno agli altri peer e ad un nodo speciale, il nodo leader; sarà il leader che, raggiunto un congruo numero di transazioni e/o intervallo di tempo, le inserirà in un blocco, ordinate in ordine temporale, e lo propagherà in rete; i nodi lo immagazzineranno localmente. Dopodiché trasmetteranno ai peer il blocco e se il blocco immagazzinato coinciderà con quello ricevuto da almeno i 2/3 degli altri nodi, aggiungeranno il blocco alla catena. Questo ultimo test sarà eseguito due volte.

In [Bessani et al., 2014] gli autori forniscono un'implementazione di un protocollo di consenso di tipo BFT, detto *BFT SMaRT* definita dagli autori sicura ed efficiente per messaggi di piccola e media taglia, per i quali la

velocità di trasferimento è mostrata essere molto buona. Inoltre includono un protocollo per memorizzare le operazioni eseguite dal singolo nodo affinché questo possa ricostruire l'ultimo stato in caso di riconfigurazione.
Nel technical whitepaper ([Hearn, 2016]) di Corda, viene proposto l'algoritmo BFT SMaRT in caso di relazioni in cui la fiducia è meno robusta.

Una variante del PBFT è *Tendermint core*; a differenza del PBFT, però, nella fase iniziale i nodi trasmetteranno le transazioni a speciali nodi validatori; inoltre il leader è a rotazione, in quanto viene cambiato dopo la creazione di ogni blocco. Qui dunque sono solo i nodi validatori e i leader che devono essere noti.

Il principale esponente di protocollo chain-based è *BChain* ([Duan et al., 2014]). Come sopra osservato in generale protocolli di questo tipo mirano a raggiungere alti flussi ma, in generale sono meno riesilienti ai guasti; per migliorare tale resilienza, in BChain viene adottato un approccio chiamato *re-chaining*, operazione a basso dispendio computazionale, mediante la quale la catena viene riordinata nel caso di sospetto di guasto. In particolare tutte le repliche, organizzate in una catena, possono sospettare ciascuna del proprio

successore; in caso di sospetto, viene mandato un messaggio di sospetto alla catena che quindi viene riordinata togliendo l'accusato e mettendo l'accusatore in una posizione tale da non poter più accusare nessuno.
In [Duan et al., 2014] gli autori mostrano che BChain supera in pretazioni i precedenti protocolli, in particolare in caso di guasti.
L'algoritmo di consenso *Sumeragi*, implementato in Hyperledger Iroha trae ispirazione dal protocollo BChain ([Hyperledger]).

Hyperledger Indy invece utilizza il protocollo *Redundant Bizantine Fault Tolerance* (RBFT), introdotto in [Aublin et al., 2013]: gli autori , nell'intento di migliorare gli esistenti protocolli di tipo BFT, mostrano che essi non forniscono accettabili performance al verificarsi dei guasti in quanto, mirando a raggiungere alti flussi, usano una speciale replica, detta primaria, che indica alle altre l'ordine in cui le richieste devono essere processate; il problema è che tale replica può essere *smartly malicious* e quindi intaccare le performance del sistema senza che le repliche corrette se ne accorgano.
Per ovviare a tale problema, il protocollo RBFT esegue in parallelo più istanze dello stesso protocollo BFT ciascuna delle quali avente una replica primaria eseguita su differenti macchine. Tutte le istanze ordinano le richieste, ma

solo quelle ordinate da una di esse, detta istanza master, sono eseguite. Le performance delle altre istanze (dette backup instance) sono monitorate in modo da controllare che le performance dell'istanza master siano adeguate; in caso contrario (e cioè se l'istanza master è più lenta), la replica primaria dell'istanza master viene considerata malicious e quindi viene rimpiazzata.

Per completezza inseriamo in questa trattazione il *Ripple protocol consensus algorithm* (RPCA) usato nella piattaforma *Ripple*, anche se in essa non viene utilizzata una blockchain ma solo un ledger. Essa adotta una trust assumption *flessibile*: in tale protocollo il ledger è aggiornato da speciali nodi detti server, ciascuno dei quali ha una propria lista di nodi, detta *unique node list* (UNL), di cui fidarsi. I nodi trasmettono le transazioni ai server, che aggregheranno quelle valide nel *candidate set* della propra UNL; se almeno l'80% dei server della UNL giudicheranno valida una transazione candidata essa sarà aggiunta al ledger, altrimenti sarà scartata. In [Schwartz et al., 2017] si afferma che i nodi avversari tollerati sono $f < \frac{n}{5}$. Chiaramente, affinché non ci sia un fork è ovvio che ci debba essere un'intersezione non vuota tra ogni coppia di distinte UNL; in [Armknecht et al., 2015] viene provato che tale intersezione deve contenere almeno i due quinti del

numero di nodi presenti nella lista più numerosa.

1.2.4 Proof of Vote

In [Li et al., 2017] viene proposto un algoritmo di consenso, *Proof of vote* (PoV), basato su un meccanismo di voto applicabile ad una Consortium blockchain.

In tale sistema la verifica e la sottomissione dei blocchi nella blockchain è permessa solo ad alcuni enti, ma la decentralizzazione è assicurata per quanto riguarda il processo decisionale ed il rispetto delle regole: per evitare una presenza dominante all'interno del consorzio, la PoV separa il diritto di voto dal diritto di produzione dei blocchi, proprio per mantenere l'indipendenza dell'esecuzione stessa. Il ruolo dell'esecuzione della produzione di un blocco è affidato ad un team affidabile e privo di leader, reclutato attraverso l'intera rete ed eletto a rotazione.

Inoltre un unico blocco valido sarà generato dalla votazione, senza possibilità di fork ed il tempo di conferma delle transazioni è ottimizzato rispetto agli esistenti algoritmi di consenso.

Entrando nei dettagli, il consorzio è formato da compagnie che utilizzano una blockchain per condividere le informazioni e le transazioni. Un team è responsabile per la produzione dei blocchi mentre ogni blocco sarà verificato e votato da ogni compagnia. Il lavoro dei membri del team sarà supervisionato e giudicato dai membri del consorzio, in modo da allontanare membri disonesti.

Ci sono quattro tipi di nodi nella rete a seconda del ruolo che ricoprono nell'algoritmo di consenso PoV:

- commissioner;
- butler;
- butler candidate;
- ordinary user.

Un commissioner è uno dei membri del comitato della lega. Ha il potere di raccomandare, votare e valutare i butler. Inoltre ha l'obbligo di verificare e trasmettere blocchi e transazioni. Quando un blocco generato avrà ricevuto più della metà dei voti sarà considerato valido ed aggiunto definitivamente

alla blockchain.

I butler producono i blocchi a turno, quando nominati per farlo, e la nomina avviene in maniera random. Raccolgono le transazioni, le impacchettano in un blocco e lo firmano. Sono pagati in proporzione al carico di lavoro svolto. Sono l'analogo dei miner in Bitcoin. Diventano butler dopo essere stati butler candidate ed aver vinto l'elezione. I commissioner votano i butler candidate per eleggere i butler. Il numero dei butler è limitato.

Per diventare un butler candidate basta avere un account nella blockchain del consorzio, candidarsi, sottomettere una lettera di raccomandazione firmata con la chiave privata da almeno un commissioner e sottomettere un deposito; se un candidato perde un'elezione può aspettare la successiva.

Un ordinary user può arbitrariamente unirsi alla rete o abbandonarla e può solo partecipare al processo di trasmissione dei messaggi e dei blocchi.

In ogni ciclo di incarico (dei butler) vengono generati un certo numero di blocchi di cui l'ultimo contiene le informazioni riguardanti l' elezione. Dopo

la generazione di ogni blocco viene generato un numero casuale che individua il butler nominato per generare il prossimo blocco. Quando questo sarà generato, sarà ritenuto valido solo se avrà la firma di più della metà dei nodi commissioner, i quali lo firmano solo se sono d'accordo riguardo alla sua produzione. Quando un blocco è ritenuto valido è aggiunto definitivamente alla blockchain ed in quel momento tutti i butler cancelleranno le transazioni illegali dalla loro transaction pool. Dopo la generazione dell'ultimo blocco di ogni ciclo di incarico i butler in carica ed i butler candidate si candidano per il prossimo ciclo di incarico, i commissioner votano ed i risultati dell'elezione vengono scritti in uno speciale blocco; in seguito alla votazione vengono eletti i butler in carica per il prossimo ciclo.

Gli autori dimostrano che sotto l'assunzione che almeno la metà aumentata di uno dei nodi commissioner non sia avversaria, la PoV garantisce sicurezza, assenza di biforcazioni e alte prestazioni in termini di bassa latenza della transazioni.

1.2.5 Forme ibride di algoritmi di consenso

Passiamo ora ad esaminare alcuni protocolli che utilizzano sia la PoW che la PoS.

Il primo è stato introdotto in [King et al, 2012] ed è utilizzato in PPCoin (anche se in [Benton et al., 2016] questo protocollo viene definito di tipo PoS puro). Per poter appendere il proprio blocco alla catena il miner deve scegliere un output ricevuto tra gli ultimi 30 e 90 giorni e formare un struttura, detta *kernel*, contenente l'output scelto, il tempo corrente, ed un blocco di bit casuali periodicamente ricalcolato (*nStakeModifier*); in seguito deve effettuare una PoW sull'hash del kernel in modo da verificare una certa condizione che è funzione non solo della difficoltà come nella PoW ma anche del *coin age* dell'output scelto (cioè il prodotto tra il valore dell'output ed il tempo di possesso): maggiore è la coin age e maggiore sarà il target, rendendo più semplice il lavoro di mining. Se il miner riuscirà ad effettuare il mining sul blocco, spenderà l'output scelto in una transazione chiamata coinbase, creata dal miner a se stesso e precedentemente inserita nel blocco, ma la coin age di tale output sarà resettata a 0. Notiamo che il limite temporale degli ultimi 30 e 90 giorni è imposto per evitare che l'anzianità aumenti in modo da far

dominare alcuni utenti sugli altri.

Una debolezza del protocollo appena descritto è data dal fatto che i nodi potrebbero essere tentati di non partecipare al sistema di verifica per accumulare più coin age. Per risolvere questo problema, il protocollo precedente è stato modificato in [Vasin, 2014] utilizzando il puro valore di stake al posto della coin age (ed adottato in Blackcoin), ed in [Ren, 2014] (adottato in reddcoin) abbinando la coin age con una funzione esponenziale di decadimento. In [Benton et al., 2016] gli autori evidenziano anche altri problemi del protocollo di PPcoin (tra cui rational fork e bribe attack) e per migliorarlo propongono il già descritto protocollo CoA.

Anche la Proof of Activity (POA, [Benton et al., 2014]) utilizza sia la PoS che la PoW, e nasce per migliorare il protocollo di Bitcoin. Nella PoA ogni miner crea un blocco contenente il previous block header, l'indirizzo pubblico del miner, l'altezza relativa al genesis block ed un nonce, ma nessuna transazione; se poi il miner riesce a fare in modo che l'hash dell'header del blocco creato in tal modo sia più piccolo del target di difficoltà corrente, lo propaga nella rete. Dopodiché viene applicata N volte la già descritta follow

the satoshi procedure per individuare N stakeholder, con N fissato: i primi $N-1$ firmeranno l'header del blocco, mentre l'N-simo creerà il blocco da aggiungere alla catena, inserendo nel blocco vuoto un certo numero di transazioni. Il blocco sarà allora propagato nella rete e, se valido, sarà aggiunto alla blockchain.

In [Liu et al, 2017]viene proposto il *fork-free hybrid consensus with flexible Proof of Activity*, un fork free hybrid consensus che sfrutta tutti e tre i tipi di consensus precedenti: PoW, PoS e BFT. Infatti gli autori propongono dapprima una PoW generalizzata, nella quale vengono richeste soluzioni ad un problema multiplo di PoW; poi viene sfruttata la tecnica di BFT (per liberarsi da possibili fork e assicurare l'integrità delle transazioni): tali soluzioni vengono sottoposte ad una commissione (fork-free hybrid consensus, adattato da un'idea già proposta in [Pass et al., 2017]). Il flexible PoA protocol descritto dagli autori prevede poi che la commissione ruoti; la probabilità di un nodo di essere eletto nella commissione è correlata ad una opportuna funzione peso che dipende sia dalla potenza di PoW che dallo stake value del nodo. La probabilità di formazione di un blocco avversario è trascurabile se l'hash power dell'avversario (per il fork-free hybrid consensus) ed il suo peso

(per il flexible PoA) sono minori di di 1/3 del totale.

1.3 Considerazioni

Nelle sottosezioni precedenti, nel descrivere i vari algoritmi di consenso, abbiamo già evidenziato alcuni loro punti di forza e alcune problematiche, e capito per quali tipi di blockchain più si adattano.

Per esempio abbiamo già sottolineato come algoritmi di PoW siano adatti a blockchain di tipo permissionless nelle quali la totale decentralizzazione rende minore la probabilità di manomissione dei dati (e quindi maggior sicurezza) rispetto a quelle di tipo permissioned. Per contro sono caratterizzati da un alto consumo energetico e dalla necessità di possedere hardware altamente specializzati (anche se alcune varianti viste richiedono sforzi computazionali minori, [Tromp, 2014]); ciò può portare alla creazione di mining pool che rappresentano un pericolo di attacco al consenso in caso di alta concentrazione di potenza di hashing (anche se in [Eyal et al., 2014, Miller et al., 2015] vengono proposti algoritmi per prevenirle). Hardware specializzati servono anche nella variante di Primecoin ideata da [King, 2013], certamente utile

per la ricerca in teoria dei numeri, e verso la quale obiettiamo che il test di Fermat non è un vero e proprio test di primalità: il piccolo teorema di Fermat garantisce che se p è primo, allora ogni intero n coprimo con p soddisfa $n^p \equiv n(mod\ p)$, ma non garantisce che se un intero soddisfa una tale congruenza sia primo.

Vogliamo ora sottolineare come la performance di algoritmi di tipo PoW è intrinsecamente limitata, indipendentemente dai vari tipi di implementazione. I due principali parametri legati alla performance sono il block size e il block frequency. Aumentare il block size per cercare di aumentare il flusso delle transazioni propagate aumenta di fatto la latenza, per il fatto che blocchi più pesanti si propagheranno più lentamente nella rete, e la latenza causerebbe problemi di sicurezza, in quanto renderebbe più probabile un fork; analogamente, anche aumentare il block frequency abbasserebbe il livello di sicurezza ([Sompolinsky et al, 2015]).

Per contro, moderni protocolli di tipo BFT hanno confermato di sostenere decine di migliaia di transazioni al secondo ed una latenza nella rete praticamente nulla ([Bessani et al., 2014]); quindi per essi la performance è

migliore rispetto ai protocolli di tipo PoW: in Bitcoin il flusso è di circa 7 transazioni al secondo (senza contare che in media si deve aspettare un'ora prima che tale transazione sia seguita da 6 blocchi, come raccomandato per considerarla sicura). Di contro la scalabilità rispetto al numero di nodi è ottima in protocolli PoW e notoriamente limitata in quelli di tipo BFT (Teorema CAP, [Brewer, 2000]).

I protocolli di tipo BFT sono considerati adatti specialmente a blockchain di tipo permissioned in quanto il loro utilizzo in blockchain di tipo permissionless esporrebbe a problemi di sicurezza dovuti ai cosiddetti Sybil attack, nei quali un attacker potrebbe partecipare al protocollo con più identità e quindi manovrare il consenso a proprio vantaggio. La necessità di conoscere l'identità di ogni nodo che partecipa al consenso può essere vista come uno svantaggio rispetto a protocolli di tipo PoW, ma in molte applicazioni delle blockchain che vanno oltre la criptovaluta (per esempio blockchain che registrano la proprietà di beni immobiliari e di terreni, o blockchain utilizzate in Supply Chain Management) è ovvio che solo nodi autorizzati devono poter gestire le transazioni.

Come già osservato nelle sottosezioni precedenti, mentre in PoW la potenza tollerata avversaria è minore del 25% della potenza di hashing ([Eyal et al., 2014]), in BFT i nodi faulty votanti tollerati devono essere meno del $33,\overline{3}\%$ del totale dei nodi votanti; inoltre, sotto tale trust assumption, è più difficile una minaccia di pericolo in protocolli di tipo BFT, in quanto per come essi sono concepiti si ha un'immediata sicurezza della definitiva inclusione di una transazione nella blockchain.

I protocolli di tipo PoS sono nati per ovviare ai problemi della PoW legati al dispendio energetico e alla necessità di possedere hardware specializzati (sottosezione 1.2.2): se nella PoW la probabilità di minare un blocco è propozionale alla potenza computazionale posseduta, nella PoS la probabilità di aggiungere un blocco alla blockchain è proporzionale alla quantità di stake. Anche se come abbiamo osservato si potrebbe pensare a generalizzare il concetto di stake ed adattare alcuni algoritmi di tipo PoS al di là delle criptovalute, in generale tali algoritmi sono utilizzati in blockchain di tipo permissionless 1.0 (secondo la definizione di [Swan, 2015]); ha quindi senso confrontarli con algoritmi di tipo PoW piuttosto che con algoritmi di tipo BFT.
In generale in protocolli PoS la performance è maggiore rispetto a quelli di

tipo PoW, e la minaccia di attacchi al consenso è minore. Il confronto però dipende dal numero di nodi della rete: in [Seang et al,2018] viene analizzata e confrontata la resilienza agli attacchi di protocolli PoW e PoS applicati in criptovalute, ed il risultato favorisce protocolli di tipo PoS in caso di pochi nodi, di tipo PoW in caso di molti nodi. Infatti dall'analisi risulta che se gli utenti sono pochi, attacchi a protocolli PoS sono possibili ma limitati e facilmente controllabili, mentre con pochi nodi aumenta il rischio di 51% attack dovuti alla selfish mining strategy in protocolli PoW.

Nella tabella Tab.1 vengono riassunte le principali differenze tra algoritmi di tipo PoW, PoS e BFT sopra descritte con maggiori dettagli.

	PoW	PoS	BFT
identità dei nodi	non nota	non nota	nota
tipo blockchain consigliato	permissionless tanti nodi	permissionless pochi nodi	permissioned
decentralizzazione	totale	alta	parziale
risarmio energetico	no	si	si
potenza avversaria tollerata	$< 25\%$ potenza computazionale	$< 51\%$ stake	$< 33,\overline{3}\%$ nodi faulty votanti
minaccia di pericolo	seria con pochi nodi	meno seria con pochi nodi	difficile
performance	limitata	buona	eccellente
esempio	Bitcoin	Nextcoin	Hyperledger Fabric
tempo necessario per un blocco nell'esempio	10 min	60 s	1 s

Tab. 1

Quanto alle differenti varianti di algoritmi esaminati nelle sottosezioni precedenti, a nostro parere meritano nota la Greedy Heaviest-Observed Sub-Tree (GHOST,[Ghost, 2015]) in ambito PoW, mentre il protocollo Chains of Activity ([Benton et al., 2016]) è sato il punto di partenza per molte varianti di tipo PoS; tra queste il protocollo Ouroboros ([Kiayias et al., 2017]) spicca per il fatto che gli autori forniscono argomentazioni formali per dimostrarne la sicurezza. Tra le forme ibride il fork-free hybrid consensus with flexible Proof of Activity ([Liu et al, 2017]) parte dalla Proof of Activity di [Benton et al., 2016] ma inserisce anche elementi di tipo BFT, sfruttando tutti e tre i tipi di consensus algorithm visti.

Degno di nota infine l'algoritmo di PoV che, applicato alle consortium blockchain, garantisce assenza di biforcazioni, sicurezza e bassa latenza delle transazioni. Ne illustreremo un'applicazione nelle Sezioni 2.3 e 2.4.

Capitolo 2

Blockchain e simulazione

2.1 Blockchain e Interoperabile Modelling & Simulation

La tecnologia blockchain permette di superare un problema della sicurezza nei sistemi distribuiti come ad esempio la modifica di dati preesistenti o l' inserimento di dati aggiuntivi da utenti non autorizzati, in quanto permette la convalida dei dati in un database condiviso senza un sistema centralizzato dedicato ([Swanson, 2015]) assicurando l'integrità dei dati stessi ([Liu et al, 2017]).

Potrebbe essere utile utilizzare tale tecnologia in simulazioni interoperabili distribuite (per esempio basate su *High Level Architecture HLA*) per confermare in modo più formale le operazioni sugli oggetti, come per esempio una loro modifica o un loro trasferimento di proprietà; applicando la tecnologia blockchain a tali simulazioni si eviterebbe che un partecipante possa fare ciò che vuole su un oggetto e si eviterebbero operazioni errate. Inoltre, per riprodurre in modo ancor più efficiente il comportamento dei partecipanti al sistema distribuito da simulare, può essere utile introdurre agenti intelligenti (IA).

Per esempio, nel caso di una supply chain, e quindi di un sistema per sua natura distribuito e con molti parametri stocastici ed alto numero di interazioni, un approccio di Modeling and Simulation (M&S) è un ben noto strumento di aiuto alle decisioni al fine di migliorare la gestione, permettendo di introdurre possibili effetti stocastici ([Bruzzone et al., 2005]); inoltre la sopra citata HLA permette di connettere diversi componenti del simulatore e di fornire una realistica rappresentazione del sistema, ancora più efficiente se combinata con l'introduzione di agenti intelligenti che meglio riprodurrebbero il comportamento dei differenti partecipanti alla supply chain: clienti, fornitori, trasportatori potrebbero essere rappresentati da agenti intelligenti

con corrispondente logica di operazione ed un all-seeing viewer potrebbe individuare eventuali criticità e suggerire come migliorare il sistema.
In blockchain è possibile salvare tutti i dati confermati relativi ad un oggetto, come il suo trasporto, cambio di proprietà, o una qualsiasi modifica.

Un altro possibile uso combinato di blockchain e simulazione riguarda la simulazione volta ad ottimizzare un sistema sulla base di input noti e con parametri modificabili: i simulatori potrebbero qui lavorare in parallelo e condividere tramite la rete blockchain quella che per ognuno è la soluzione migliore al problema, unitamente ai valori dei parametri che la realizzano; quando una possibile soluzione di ottimizzazione è propagata, gli altri simulatori possono verificarla utilizzando i relativi parametri e se l'algoritmo di consenso utilizzato la elegge come migliore essa viene inserita in blockchain.

Attualmente poco è noto sulla previsione del comportamento di sistemi basati sulla tecnologia blockchain.
Spesso la simulazione dell'intero sistema distribuito potrebbe essere troppo impegnativa e quindi richiederebbe sforzi comparabili a quelli richiesti dall'implementazione del sistema finale: quindi è conveniente concentrarsi solo

su un particolare aspetto critico della tecnologia da simulare.

Per esempio, una limitazione della tecnologia blockchain rispetto all'adozione di un sistema tradizionale è data dalla latenza, sia nell'inserimento di una transazione in un blocco, sia dovuta al fatto che la garanzia di sicurezza si ha solo dopo che nuovi blocchi successivi sono aggiunti alla blockchain, oltre ovviamente a quella dovuta ai ritardi della rete, dal numero di transazioni in corso di elaborazione, e dal tipo di algoritmo di consenso utilizzato. Una stima della latenza di un sistema basato su tecnologia blockchain potrebbe quindi essere utile per la progettazione del sistema stesso, per capire se, nel caso d'uso, essa può essere accettabile a fronte dei potenziali benefici che l'uso della blockchain porterebbe a confronto di un sistema tradizionale, quale per esempio la fiducia decentralizzata. A tal proposito recentemente in [Yasaweerasinghelage et al, 2017] viene mostrata la possibilità di utilizzare strumenti di M&S per predire la latenza di sistemi basati su tecnologie blockchain.

Inoltre, poiché le blockchain possono essere impiegate in modi differenti, sarebbe utile testare i parametri dell'implementazione che possono influenzare la sicurezza, la facilità d'uso del sistema e la sua efficienza prima dell'implementazione basata su blockchain. In [Kreku et al., 2017] gli autori mostrano

come strumenti di simulazione possono essere utilizzati per scoprire limiti e opportunità in implementazioni basate su blockchain di sistemi IoT (*Internet of Things*).

Un altro aspetto critico delle blockchain che si potrebbe simulare per stimare il possibile rischio di corruzione dei dati è quello di un probabile attacco al consenso considerando differenti condizioni iniziali, quali per esempio il numero dei nodi. Per farlo si potrebbe utilizzare una delle implementazioni di blockchain open source disponibili per ricreare la rete ed effettuare i test richiesti.

2.2 Esempio relativo al caso di studio: previsione di domanda e offerta nell'industria e in logistica

Una supply chain è definita come la catena che lega ogni entità della produzione e il processo di fornitura dai materiali grezzi fino all'utente finale [New et al., 1995, Scott et al., 1991].

Una supply chain dunque comprende molti sistemi, legati alla produzione, al deposito, al trasporto e alla distribuzione.

Se l'intero sistema non è coordinato, cioè se ogni entità pensa alla propria ottimizzazione locale senza considerare l'impatto delle proprie decisioni sulle altre, ciò può portare ad una maggiore variazione tra la domanda e l'offerta a livello globale; viceversa è noto come la riduzione dei costi globali e il successivo impiego dei conseguenti risparmi tra le entità sia il migliore interesse perseguibile [Lee et al., 1993, Lee et al., 1999, Simchi-Levi et al., 2000, Umeda et al, 1998].

Per una ottimizzazione globale è necessaria coordinazione tra le varie entità; la coordinazione inoltre apporta non solo ottimizzazione, costo globale minore e minore variazione tra domanda e offerta, ma anche un migliore flusso dei materiali e una diminizione dei tempi di consegna.

Per la coordinazione è necessaria la comunicazione tra le varie entità, e tale comunicazione deve essere approfondita, senza però portare a troppa perdita di tempo, oltre ovviamente ad essere basata su informazioni *accessibili e trasparenti*, [Lee et al., 1997].

La mancanza di informazione tra i livelli di una supply chain è stata individuata da Forrester e da Sterman, rispettivamente in [Forrester, 1961,

Sterman et al, 1992], come una delle cause dell' *effetto Bullwhip*, o effetto Forrester, definito da Forrester stesso come l'amplificazione della variabilità del segnale di domanda / ordine che si riscontra man mano che questo risale, da valle a monte, dal retailer al manufactorer, lungo la filiera logistica.
In [Lee et al., 1997] gli autori aggiungono tra le cause dell'effetto Bullwhip gli errori nella previsione della domanda.

Molti autori hanno studiato come migliorare l'accuratezza nella previsione della domanda, spesso evidenziando come ciò possa tamponare l'effetto Bullwhip.
Molte ricerche però si sono focalizzate sulla previsione della domanda di un singolo livello della catena; in [Kimbrough et al, 2002, McBurney et al., 2002, Chen et al., 2000] gli autori hanno considerato i vari livelli della catena per la previsione della domanda, ma hanno assunto che ogni livello adottasse lo stesso sistema di gestione dell'inventario, cosa che nella realtà succede di rado.
In [Chen et al., 2000] gli autori hanno quantificato l'effetto Bullwhip quando sono utilizzati i modelli di previsione moving-average ed exponential- smoothing; in [Zhao et al., 1993] gli autori confrontano i modelli di previsione

double exponential smoothing e Winter's exponential, mostrando che quest'ultimo produce una minore deviazione standard dell'errore di previsione ma aumenta il costo totale e crea instabilità nella pianificazione. Inoltre tutti i metodi precedenti sono basati su molti dati precedenti e danno una previsione della domanda di mercato nel periodo $t+1$ senza tener conto del costo totale della catena nel periodo precedente t.

In [Liang et al., 2006] gli autori utilizzano il *Real-coded Genetic Algorithm* (RGA) [Michalewicz, 1996] in quanto non richiede grandi quantità di dati e poiché la sua funzione obiettivo è il costo minimo totale: nel codice genetico ogni bit rappresenta un ordine per ognuna entità, quindi ogni ordine di ciascuna entità è considerato durante il processo di evoluzione, e il costo totale della supply chain è ottimizzato globalmente. Inoltre il modello di supply chain permette di usare nei vari livelli diversi sistemi di gestione dell'inventario.

Il modello agent based permette agli agenti di comunicare e di condividere con le altre entità le informazioni sulla domanda e l'assunzione fatta dagli autori è che ogni entità sia disposta a condividere in modo onesto le informazioni con le altre; sotto tale assunzione il costo totale è minimizzato. In tale

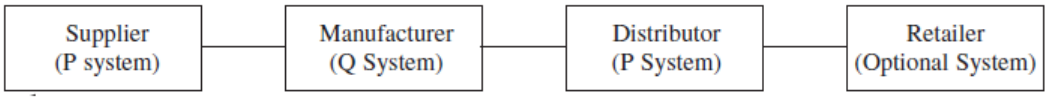

Figura 2.1: Modello di supply chain

contesto è evidente come l'applicazione della tecnologia blockchain potrebbe rendere fondata tale assunzione.

Passiamo ora ad esaminare in dettaglio il lavoro svolto in [Liang et al., 2006]. Il modello proposto si basa sul *beer game* [Sterman, 1988] a 4 livelli (fornitore, produttore, distributore, venditore, Figura 2.1 da [Liang et al., 2006]). Si suppone una singola entità per ogni livello, e si considera una politica di inventario chiamata (P Q P O), cioè si suppone che il fornitore e distributore adottino un sistema di revisione periodico del magazzino (P), il produttore un sistema di revisione continuo (Q) e il venditore un sistema opzionale (O).

I tre sistemi sopraelencati sono in generale i tre tipi possibili di sistemi di inventario indipendenti dalla domanda [Krajewski et al, 2002].

In un sistema di revisione periodico lo stato del magazzino viene revisionato ogni periodo di tempo fissato ed in ogni revisione (se necessario) viene effettuato un ordine in modo da raggiungere un fissato livello di stock, mentre

in un sistema di revisione continuo il deposito rimanente di un oggetto è monitorato costantemente in modo da determinare quando è necessario un riordine, quindi il tempo tra gli ordini è variabile. Il sistema opzionale è un misto dei precedenti, in quanto il tempo di revisione è fissato come in quello periodico, ma un riordine è effettuato solo se si è raggiunto un livello minimo in magazzino, come in un sistema continuo.

La Tabella seguente riassume le principali differenze tra i tre tipi sopraelencati di sistemi di inventario.

	Q system	O system	P system
revisione	continua	periodica	periodica
tempo tra un ordine e l'altro	variabile	variabile	fisso (dopo ogni revisione)
quantità di ordine	fissa	variabile	variabile

Tab. 2

In generale un sistema continuo sarà conveniente nel caso di spazi per il deposito limitati, alti costi di stoccaggio, costi di revisione di gestione e di

trasporto non significativi, mentre è alta la necessità di fare l'inventario.
Al contrario un sistema periodico sarà conveniente nel caso in cui non ci sono problemi di spazi per il deposito, la necessità di fare l'inventario non è alta, i costi relativi allo stoccaggio non sono significativi, mentre è auspicabile una diminuzione dei costi relativi alla revisione del magazzino e alla gestione.
Essendo il sistema opzionale un misto tra i due precedenti, evita revisioni continue e acquisti non necessari, quindi è conveniente quando i costi di revisione del magazzino e di ordine sono significativi.
Le osservazioni precedenti avvalorano le scelte degli autori che, come sopra menzionato, suppongono che fornitore e distributore adottino un sistema di revisione periodico del magazzino, il produttore un sistema di revisione continuo e il venditore un sistema opzionale.

Ogni livello, nel modello agent-based proposto in [Liang et al., 2006], ha un *control agent* ed un *demand forecast agent* (Figura 2.2 da [Liang et al., 2006]).

Il control agent accumula dai manager della supply chain dati storici relativi alla domanda e le strategie adottate dai manager stessi, al fine di costruire una regola di base per la gestione della supply chain, mentre il demand fore-

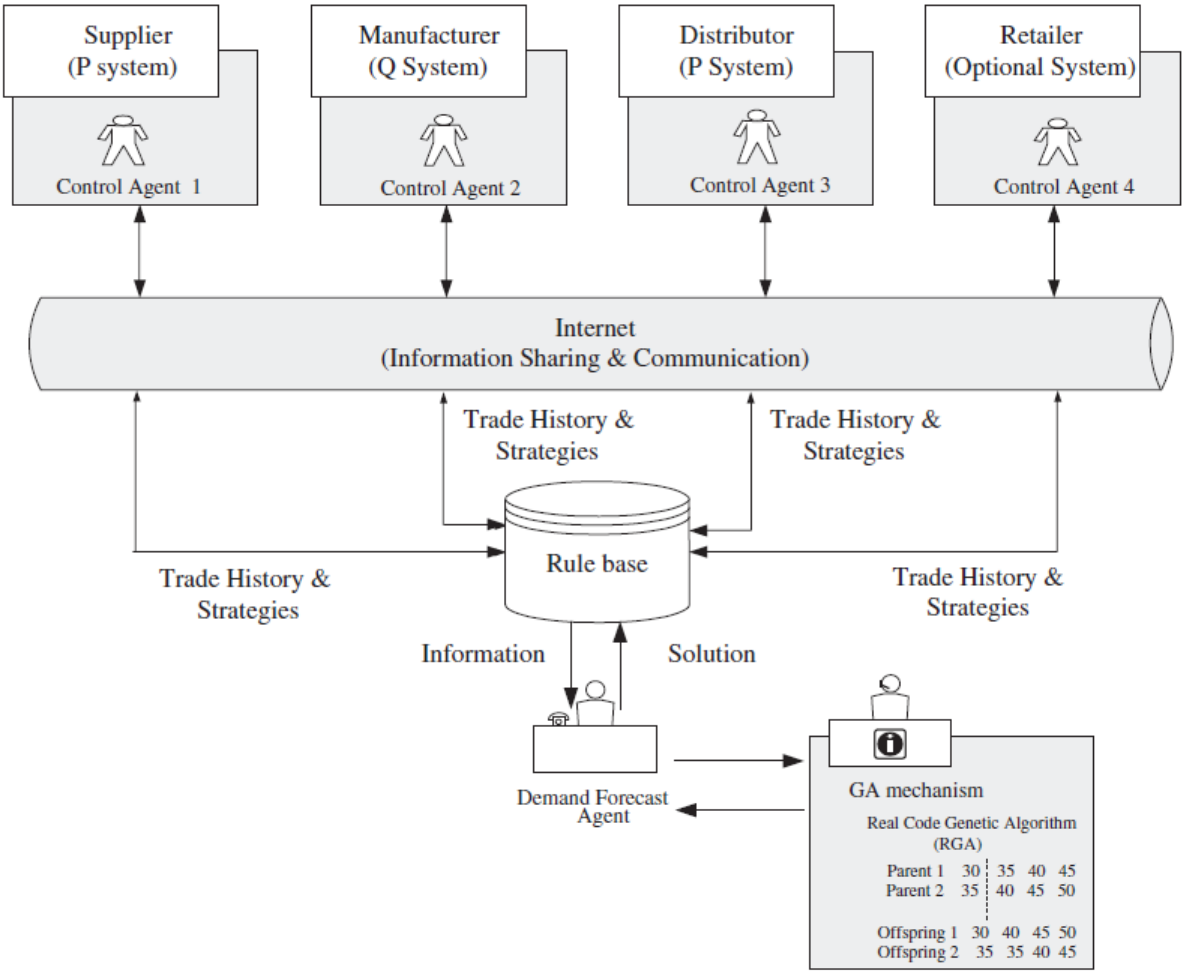

Figura 2.2: Architettura del sistema

scast agent trasforma l'esperienza dei manager e fornisce un meccanismo di previsione teso a minimizzare il costo totale della supply chain; la previsione della soluzione ottimale che stabilisce in tale ottica la quantità di ordini per ogni livello è trovata grazie all'utilizzo del sopracitato RGA.
In particolare, i control agent si scambiano informazioni tra stato della produzione, costi della produzione e di trasporto, costi di magazzino, livelli di magazzino, quantità degli ordini e domanda del cliente.
Dopodiché viene creata una *demand rule table* contenente informazioni che condizionano il crossover range quando viene calcolata la quantità di domanda; per esempio, in un sistema di revisione continua, la demand rule table conterrà per ogni oggetto, il numero di unità disponibili on-hand in magazzino (OH), il numero di oggetti la cui ricezione è programmata ma non ancora avvenuta (SR, da scheduled receipts, ordini aperti), il numero di unità backordered (BO), cioè ordinate ma il cui ordine non è ancora stato evaso ed il lead time (L) relativo a quel livello. Inoltre nella tabella vengono inseriti i crossover range, in base all'esperienza dei manager.

Il demand forecast agent applica la *Rought Set Theory*, teorizzata da Z. Pawlak nel 1982 e sviluppata dallo stesso autore in [Pawlak, 1991] con lo

Rule	L	OH	SR	BO	Crossover range
1	0	20	10	0	[0–20]
2	0	10	15	0	[10–15]
3	1	20	15	0	[10–15]
⋮	⋮	⋮	⋮	⋮	⋮

Figura 2.3: information table

scopo di classificare oggetti utilizzando informazioni imprecise o incomplete, alla demand rule table che rappresenta l'information table alla quale applicare la teoria.

Un'information system nella rought set theory è una coppia $I = (U, A)$ con $U = \{x_1, \ldots, x_n\}$ insieme finito di oggetti ed $A = \{a_1, \ldots, a_m\}$ insieme finito di attributi. Ad I si associa la tabella di informazione (information table) che in sostanza può essere vista come una matrice T_{ij} di n righe corrispondenti agli oggetti ed m colonne corrispondenti agli attributi, costruita in modo che T_{ij} è il valore dell'attributo a_j relativamente all'oggetto x_i.
Nel nostro esempio riguardante un sistema di revisione continuo, l'information table attribuisce per ogni oggetto il valore dell'attributo L, OH, SR BO e Crossover range come nella Figura 2.3 (da [Liang et al., 2006]).

Dopodiché viene effettuata una partizione dell'insieme degli attributi $A = C \sqcup D$, in modo da dividerli in Decisionali e Condizionali. Nell'esempio $C = \{L, OH, SR, BO\}$ mentre $\{Crossover\ range\} = D$.

Dato un sottoinsieme di attributi $P = \{a_{j_1}, \ldots, a_{j_r}\}$, possiamo definire una relazione di equivalenza su U, $\sim_P$ in modo che

$$x_i \sim_P x_l \Longleftrightarrow T_{ij_h} = T_{lj_h}\ \forall\ h = 1, \ldots, r,$$

cioè due oggetti appartengono alla stessa classe di equivalenza se hanno gli stessi valori per quanto riguarda gli attributi di P.

Le classi di equivalenza ottenute sono dette insiemi elementari; per ogni oggetto $x_i \in U$, si denota

$$[x_i]_P = \{x_l \in U \mid x_i \sim_P x_l\}.$$

Sia ora $X \subset U$ qualunque e $P = \{a_{j_1}, \ldots, a_{j_r}\} \subset A$.
Si definisce approssimazione inferiore di X rispetto a P l'insieme

$$Inf_P(X) = \{x \in U \mid [x]_P \subset X\};$$

l' approssimazione superiore di X rispetto a P è invece definita da:

$$Sup_P(X) = \cup_{x \in X}[x]_P$$

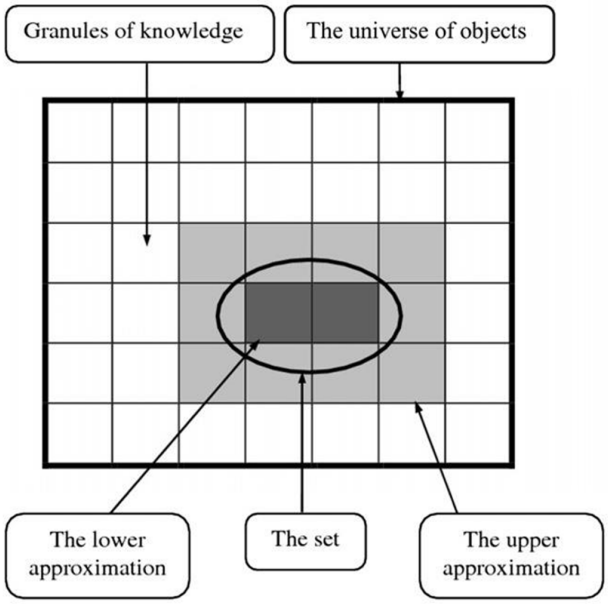

Figura 2.4: Approssimazione inferiore e superiore di un insieme rispetto a P

(Figura 2.4, da [Pawlak, 1991]). Ovviamente si ha:

$$Inf_P(X) \subseteq X \subseteq Sup_P(X); \tag{2.1}$$

inoltre se X è unione di insiemi elementari allora vale

$$Inf_P(X) = X = Sup_P(X)$$

e l'insieme è detto *crisp* (insieme esatto), mentre se nella (2.1) i contenimenti sono stretti l'insieme è detto *rough* (approssimato).

Applicando quanto sopra definito ad ogni classe di decisione elementare, cioè ad ogni classe di equivalenza di U rispetto a D, si approssima ogni classe di decisione elementare $[x_i]_D$ rispetto agli attributi condizionali, cioè rispetto a $P = C$; in particolare

$$Inf_C([x_i]_D) = \{x_l \in U \mid [x_l]_C \subset [x_i]_D\}$$

definisce l'insieme delle regole certe, in quanto per tutti gli oggetti che verificano le stesse condizioni è presa la stessa decisione.

Si passa poi alla riduzione degli attributi condizionali, con lo scopo di eliminare quelli superflui: si cerca un sottoinsieme minimale $C' \subset C$ nel senso che un'ulteriore eliminazione degli attributi da esso minerebbe le informazioni originali.

Object no.	F_1	F_2	F_3	F_4	Crossover range
1	0	20	X	X	[0–20]
	X	20	10	X	[0–20]
2	X	10	15	0	[10–15]
	0	X	15	0	[10–15]
	0	10	X	0	[10–15]
	0	10	15	X	[10–15]
3	X	X	15	0	[10–15]
4	X	X	X	5	[10–20]
	X	X	15	5	[10–20]
5	X	25	X	X	[0–10]
6	X	X	X	5	[10–20]
	X	X	15	5	[10–20]
7	X	X	20	X	[10–15]
	X	15	X	0	[10–15]
8	0	20	X	X	[0–20]
	X	20	10	X	[0–20]
9	X	X	15	0	[10–15]
10	X	X	X	5	[10–20]

Figura 2.5: Reduct rule table

In [Pawlak, 1991] viene descritta la procedura utilizzata in [Liang et al., 2006] per costruire la decision table, che nell'esempio è una tabella (denominata reduct rule table in [Liang et al., 2006], Figura 2.5 da [Liang et al., 2006]), cioè una tabella da leggersi in un'ottica di "$if, \cdots, then$": per ogni oggetto corrispondente alle righe, si pone la condizione if davanti ad ogni attributo condizionale e si pone then davanti a quelli decisonali.

Le regole generate dall'approccio rough set serviranno a vincolare il valore del crossover range che sarà ora dinamicamente calcolato mediante l'utilizzo

di un algoritmo RGA, basato sulla rappresentazione in virgola mobile e con due operazioni genetiche: il crossover e la mutazione.

Un cromosoma è rappresentato da 4 geni, cioè da una 4-upla corrispondente rispettivamente al demand order per fornitore, produttore, distributore e venditore; la fitness function, che determina il miglior cromosoma in ogni generazione e determina quando terminare l'evoluzione, è la funzione costo totale della supply chain, cioè la funzione che somma i costi di mantenimento del magazzino, del trasporto, degli ordini per ogni livello e per tutto il periodo preso in considerazione.

Due genitori $(x_1, \cdots, x_4)$ e $(y_1, \cdots, y_4)$ producono un figlio $(z_1, \cdots, z_4)$ con $z_i = x_i$ o $z_i = y_i$ con uguale probabilità (crossover probability) e il secondo figlio è creato invertendo le entrate, come mostrato nell'esemio riportato nella Tabella 3:

Genitore 1	30	35	40	45
Genitore 2	35	40	45	50
Figlio 1	30	40	45	50
Figlio 2	35	35	40	45

Tab. 3

La mutazione è definita in modo tradizionale: le entrate di un cromosoma hanno la stessa probabilità di essere oggetto della mutazione e se a mutare è l'entrata x_k, con $x_k \in [a_k, b_k]$, il valore ottenuto dalla mutazione $x_k^{'}$ è generato dalla distribuzione uniforme $U(a_k, b_k)$.

Il demand forecast agent applica l'RGA per calcolare la quantità di ordine ottimale per ogni livello: viene generata una popolazione di cardinalità definita dall'utente, generando in maniera casuale ogni gene da un fissato range (che stabilisce la massima quantità di ordine ammessa per ogni livello) ed ogni membro della popolazione viene valutato rispetto alla fitness function.
Viene poi selezionata una parte di popolazione mediante il *roulette wheel method* [Holland, 1992] e viene applicata l'operazione di crossover sopra descritta ad ogni coppia di individui della popolazione selezionata in modo da generare un figlio, ma nel rispetto dei limiti imposti dal crossover range ottenuti nella decision table; i due figli ottenuti come sopra descritto sono aggiunti alla nuova popolazione.
Successivamente si passa all'operazione di mutazione dei figli e si crea una nuova popolazione aggiungendo a quella attuale gli individui ottenuti da tale operazione.
Si ripete quanto detto sopra finché il tempo è finito, la popolazione converge o nessun cromosoma migliore è trovato.
Il cromosoma finale rappresenta la quantità di ordine ottimale per ogni livello.

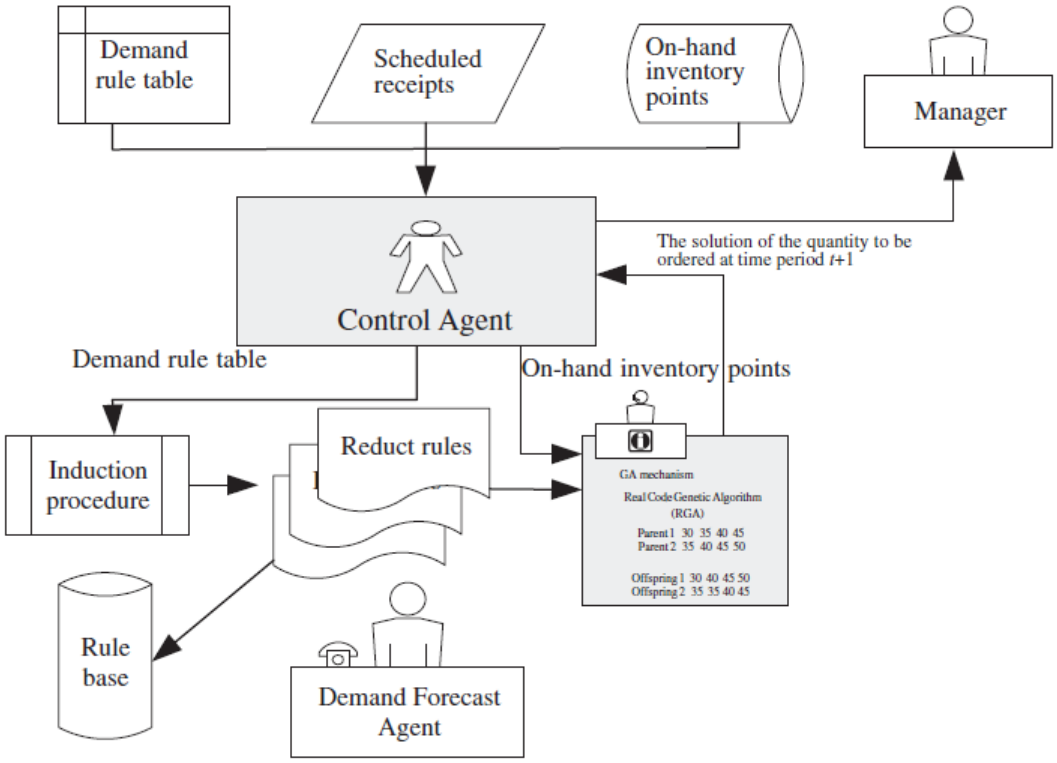

Figura 2.6: Processo di previsione della domanda

Dopodiché il demand forecast agent condivide la soluzione trovata con i control agent.

L'intero processo è illustrato nella Figura 2.6 (da [Liang et al., 2006]).

In [Liang et al., 2006] è dunque descritto un rule-based system con lo scopo di prevedere la quantità di ordine in una supply chain di 4 livelli che usa

un sistema di revisione del tipo (P Q P O). La rough set theory è utilizzata per inserire l'esperienza dei manager nelle regole, regole che sono *condivise* da tutte le entità della supply chain e che vincolano l'applicazione di un algoritmo RGA applicato per trovare la quantità di ordine ottimale cercata, che ottimizza anche il costo totale della supply chain.
Come sottolineato dagli stessi autori, l'assunzione che viene fatta è che le informazioni condivise tra le varie entità siano *veritiere* e che quindi ci sia una relazione di *fiducia* tra i vari agenti; sotto tale assunzione l'ottimizzazione è migliore e l'effetto bullwhip è ridotto.
L'applicazione della tecnologia blockchain permetterebbe di rendere fondata questa assunzione, in quanto garantirebbe per sua natura la sicurezza dei dati pubblicamente condivisi tra gli agenti.

Le sfide critiche nel proteggere l'integrità di una supply chain appaiono adatte per l'adozione della tecnologia blockchain: essa è una potenziale soluzione per migliorare la sicurezza l'integrità e la provenienza dei dati oltre che per migliorare le funzionalità e ottimizzare la gestione della supply chain stessa.

Adattare la tecnologia blockchain ad una supply chain permetterebbe:

a) di identificare un prodotto in modo univoco con informazioni convalidate e verificate;

b) tracciare il prodotto consentendo a fornitori, produttori e distributori di fornire tutte le informazioni relative al tracciamento dello stesso nel ledger distribuito con automatica verifica di tali informazioni;

c) segnalare prodotti sospettati di contraffazione o pericolosi;

d) segnalare eventuali non osservanze di particolari regole di trasporto;

e) segnalare eventuali difformità da regole contrattuali condivise dalla rete di business;

f) creare un preciso modello di previsione della domanda ed ottimizzare produzione e logistica.

Come sopra anticipato, la tecnologia blockchain si adatta al controllo della provenienza di un oggetto, tuttavia, come osservato in [Bruzzone et al., 2018], i report riguardanti tale applicazione sono rari e generici, anche se molte compagnie hanno già annunciato l'intenzione di utilizzare le blockchain per il tracciamento dei propri prodotti, tra cui IBM [Galvin, 2017], De Beers

(che ha annunciato un progetto di monitoraggio dell'intera supply chain legata ai diamanti [Butcher, 2018]) e le compagnie Provenance e Agridigital [Bermingham, 2017], ma non sono ancora pubblici i relativi dettagli tecnici.

Nel seguito analizziamo come poter applicare la tecnologia blockchain ad una supply chain evidenziando come ciò permetterebbe di ottenere quanto sopra elencato.
Nel trattare in generale quali saranno gli asset, le transazioni ed i nodi di una blockchain applicata ad una supply chain ci riferiremo all'esempio descritto nella Figura 2.7, raffigurante una supply chain che da una fattoria produrrà prodotti caseari e lavorati di carne.

In una blockchain applicata ad una supply chain un *asset* sarà qualsiasi *cosa* che abbia un valore e che possa essere trasferito da un mittente ad un destinatario: nell'esempio in figura sono asset non solo le mucche, il latte o il formaggio, ma anche i *contratti*, per esempio il contratto tra fattore e azienda casearia.
Ogni asset sarà identificato dall'ID ma anche da una serie di informazioni: nell'esempio, la mucca sarà identificata dalla data di nascita, dal tipo di ali-

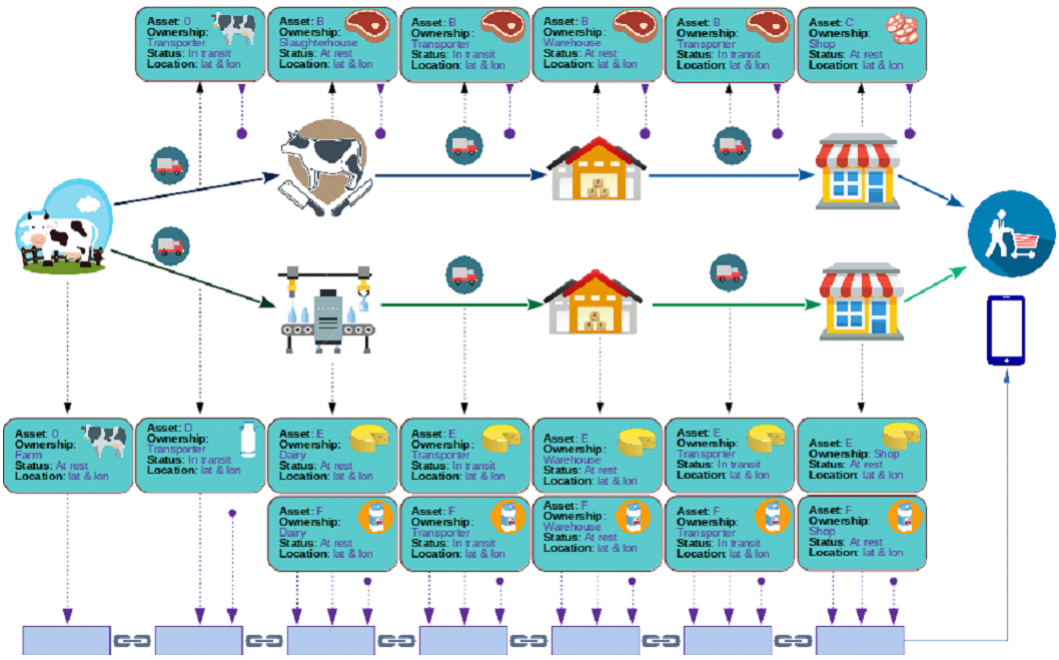

Figura 2.7: Esempio di supply chain basata su Blockchain

mentazione utilizzato, dalla fattoria di nascita e di crescita; un contratto avrà il suo ID, ma anche la data di stipula, le condizioni, la scadenza, le eventuali penalità in caso di inadempimento.
Nel caso di contratto di trasporto tra produttore e spedizioniere le condizioni potrebbero riguardare anche la temperatura durante il viaggio e l'idoneo trasporto, e prevedere una penalità in caso di inadempimento. A tal proposito, si potrebbero installare sensori nei vettori di trasporto per monitorare che la temperatura si mantenga entro il range ammissibile; tali informazioni sulla temperatura di trasporto potrebbero essere trasmesse nella blockchain e, qualora si rilevasse un superamento dei limiti stabiliti, il contratto potrebbe prevedere una penalità.

Una *transazione* della blockchain rappresenterà una qualsiasi operazione effettuata su un asset: una modifica (per esempio un' operazione che trasforma un oggetto in un altro), il trasferimento dello stesso (per esempio dal magazzino al veicolo che lo trasporta e dal veicolo al cliente destinatario), o la creazione di un oggetto da altri oggetti esistenti; il mittente e il destinatario della transazione corrisponderanno a persone, macchine o organizzazioni che possiedono e rispettivamente otterranno tale oggetto nell'operazione eseguita

su di esso.

Mittente e destinatario della transazione avranno una chiave pubblica e una privata.

Le fasi di scambio di un bene fisico tramite blockchain possono essere sintetizzate nelle seguenti (Figura 2.8): autorizzazione del venditore quale proprietario attuale, immissione di un'offerta da parte del venditore, accettazione dell'offerta da parte di un acquirente, verifica della quantità di fondi, lo scambio del bene, trasferimento di fondi dall'acquirente al proprietario e conferma della transazione da parte di venditore e acquirente.

Una transazione di una blockchain applicata ad una supply chain avverrà in modo simile, anche se il trasferimento di fondi potrebbe essere non necessario, e il ruolo di acquirente o venditore potrebbe essere rivestito anche da macchinari o altre entità.

Nell'esempio descritto nella figura, sono transazioni la spedizione della mucca (o del latte) dalla fattoria alle rispettive aziende produttrici, o la trasformazione del latte in formaggio, ma anche una variazione di un contratto.

In caso di un numero elevato di transazioni, potrebbe essere utile inserire nel database distribuito solo la transazione finale ed utilizzare al di fuori

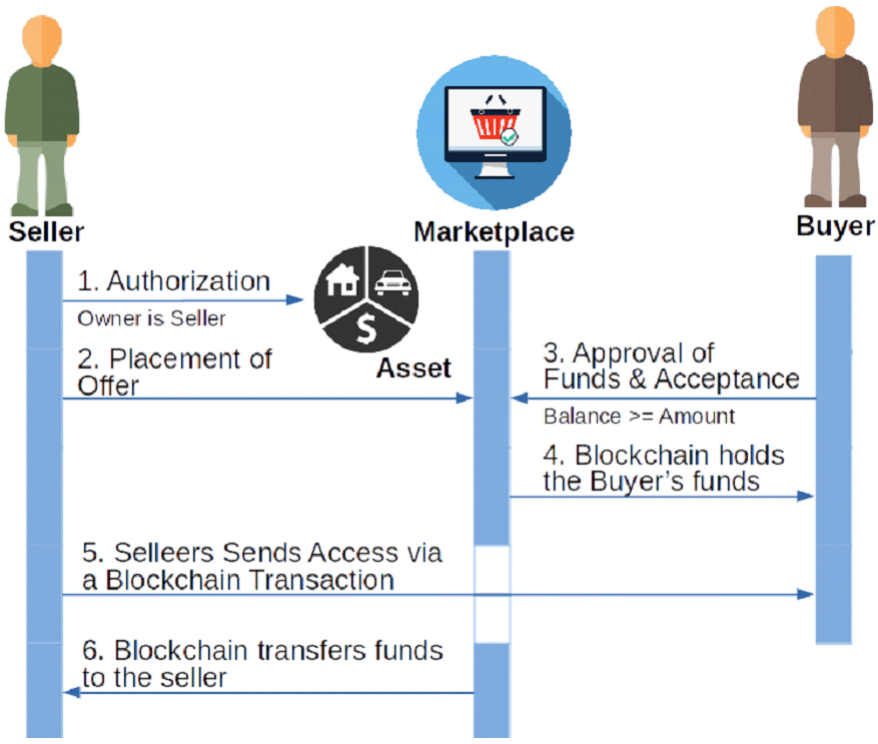

Figura 2.8: Fasi di scambio di un bene fisico

della blockchain smart contract, per esempio gli *HTLC* (*Hashed Timelock Contracts*), e le *Multisig* (*multisignature transactions*), per garantire la correttezza delle operazioni, anche se eseguite al di fuori della blockchain. A tal proposito alcuni studi si sono focalizzati sull'implementazione e miglioramento di tali tecniche, [Ramachandran et al., 2017] .

Per quanto riguarda i nodi partecipanti alla rete blockchain, essi avranno ruoli diversi e non tutti saranno validatori; inoltre non tutti avranno accesso ad ogni informazione. Quindi si ritiene che il tipo di blockchain adatta per essere abbinata ad una supply chain sia di tipo permissioned.
Nell'esempio della Figura 2.7, i nodi partecipanti saranno il fattore, le aziende casearia e produttrice di carne e derivati e gli spedizionieri; i trasportatori non potranno accedere ai dati relativi ai contratti e alle transazioni tra fattore e azienda casearia, e analogamente il fattore non avrà accesso ai dati relativi a transazioni e contratti tra azienda casearia e spedizionieri.
Se la rete di business aderisce alle regole del commercio equo solidale, nella rete blockchain potrebbe essere anche presente un nodo supervisore, autorizzato a controllare che negli asset di tipo contratto tra fattore e aziende produttrici ci sia un adeguato compenso; analogamente potrebbe essere pre-

sente un nodo corrispondente ad un consorzio di qualità per monitorare per esempio la provenienza e le condizioni del trasporto. Il fattore o lo spedizioniere potrebbere essere nodi che non possiedono copie della blockchain, ma che possono richiedere una transazione: per esempio il fattore potrebbe richiedere dal proprio smartphone una procedura di pagamento da parte dell'azienda casearia (nodo validatore) per un carico di latte, o lo spedizioniere potrebbe fare altrettanto per il trasporto di formaggio dall'azienda casearia al supermercato.
Nodi validatori potrebbero essere le aziende produttrici, il distributore e il retailer. Se poi la fattoria fosse sudamericana, potremmo avere come nodo validatore anche l'importatore, la sede centrale dell'azienda casearia e la filiale sudamericana.

Una volta stabiliti i nodi ed i loro ruoli all'interno della blockchain di tipo permissioned, il funzionamento della blockchain avverrebbe come descritto nella Sezione 1.1: ogni transazione proposta sarà verificata indipendentemente dai nodi validatori che la propagheranno a loro volta nella rete, dopo averne accertato la validità; ogni validatore poi la inserirà nel proprio blocco candidato, e l'algoritmo di consenso decreterà quale sarà il nuovo blocco da

aggiungere alla blockchain.

Come osservato nella Sottosezione 1.3, per blockchain di tipo permissioned è conveniente utilizzare algoritmi di consenso di tipo BFT, che garantirebbero anche tempi più veloci (si confronti Tab. 1); se poi la blockchain è una consortium blockchain potrebbe essere applicato il PoV.

Torniamo all'esempio del fattore che invia tramite smartphone una richiesta di pagamento per un carico di latte: i validatori accerteranno che il fattore sia autorizzato ad effettuare la transazione, che la firma digitale sia autentica, che la richiesta sia formalmente corretta ed in caso di esito positivo la propagheranno in rete e la inseriranno in un blocco candidato; poi si passerà all'applicazione dell'algoritmo di consenso scelto per l'approvazione di un nuovo blocco da aggiungere alla catena.

Alla luce di quanto detto sopra è chiaro come l'utilizzo della tecnologia blockchain potrebbe identificare ogni prodotto in modo univoco con informazioni convalidate e verificate (a) e tracciare il prodotto consentendo a fornitori, produttori e distributori di fornire tutte le informazioni relative al tracciamento dello stesso nel ledger distribuito con automatica verifica di tali informazioni. Infatti nella blockchain saranno imagazzinate tutte le prece-

denti transazioni eseguite su ogni asset, in modo tale da tenere traccia di ogni fase del ciclo vitale di un prodotto, e garantire la veridicità delle informazioni sulla provenienza dello stesso ([Kim et al, 2018], [Liang et al, 2017]): i dati marcati temporalmente riguardanti ogni fase del ciclo vitale di un prodotto possono essere immagazzinati e verificati in futuro (b).

Tale tracciamento consentirebbe anche la segnalazione di prodotti sospettati di contraffazione o pericolosi (c).

La garanzia di idoneità del trasporto è un'esigenza fondamentale in una supply chain di tipo alimentare in quanto spesso è legata alla sicurezza dei prodotti: in [Ministero della Salute, 2015] il Ministero della Salute ha redatto un manuale contenente indicazioni per un'applicazione semplificata dei principi HACCP (Hazard Analysis and Critical Points), ove viene sottolineato quanto l'idoneità del veicolo di trasporto e le condizioni in cui avviene il trasporto stesso hanno un ruolo fondamentale per tale scopo. Come sopra evidenziato nell'esempio, l'integrazione di sensori posizionati nei veicoli con la blockchain permetterebbe un sicuro controllo della uniformità alle regole di trasporto necessarie per il tipo di prodotto trasportato (d).

Per quanto riguarda il rispetto di eventuali regole contrattuali (e), l'appartenenza alla rete della blockchain da parte di opportuni supervisori assicure-

rebbe il controllo in tal senso: nel caso di adesione alle regole del commercio equo solidale un nodo supervisore potrebbe controllare i contratti in modo da garantire un adeguato compenso; nel caso di certificazione biologica un nodo potrebbe essere preposto ad un controllo costante in tal senso, e in maniera analoga la presenza di un nodo supervisore potrebbe in generale controllare il rispetto di opportune regole.

Infine l'integrazione tra la blockchain e la tecnologia RFID (*Radio Frequency Identification*), già utilizzata nel settore agroalimentare, renderebbe possibile inserire nella blockchain le informazioni relative alla rilevazione automatica a distanza degli oggetti, anche durante il loro trasporto, per un più accurato tracciamento.

Oltre agli ovvi vantaggi per il consumatore, che in tal modo potrebbe verificare l'origine dei beni acquistati nell'ultimo livello della supply chain, l'integrazione tra le tecnologie blockchain, RFID, GIS (*Geographic Information System*) e GPS (*Global Positioning System*) permetterebbe di avere costantemente l'attuale stato della supply chain; in particolare ciò permetterebbe l'acquisizione di dati statistici precisi che permetterebbero il migliormento della logistica (f).

Inoltre, l'applicazione della tecnologia blockchain ad una supply chain per-

metterebbe di avere dati sicuri riguardanti consegne e ordini, per cui permetterebbe di creare un preciso modello di previsione della domanda e dell'offerta (f).

Concludendo:

- l'applicazione della tecnologia blockchain nella simulazione di una supply chain garantirebbe una certificazione del dato della simulazione e la fiducia reciproca tra le varie entità, per esempio tra i control agent del sistema proposto in [Liang et al., 2006] per la previsione della domanda;

- viceversa, l'applicazione di una blockchain ad una supply chain reale permetterebbe l'identificazione e il tracciamento dei prodotti mediante informazioni convalidate e verificate e quindi permetterebbe la creazione di un preciso modello di previsione della domanda e l' ottimizzazione della produzione e della logistica.

  Introdurre la tecnologia blockchain all'interno di una supply chain presenta però alcune criticità: ogni componente della catena dovrebbe implementare questa funzionalità, altrimenti l'assenza di informazioni anche a livello di un solo step (legato per esempio alla trasformazione

di un oggetto) potrebbe rendere inutile l'intero sistema. Ma come descritto sopra, affinché l'utilizzo della tecnologia sia efficiente e produca i risultati sopra elencati, sarebbe necessario installare adeguate attrezzature di tracciamento che richiederebbero un investimento di tempo e denaro.

2.3 Esempio relativo al caso di studio: sicurezza sul territorio locale

Analogamente a quanto visto per le supply chain, la tecnologia blockchain potrebbe essere un utile strumento per la *prevenzione* e *gestione* dei disastri naturali.

Ciò può avvenire in due modi: da un lato si potrebbero certificare sulla blockchain tutti i dati relativi alle opere di pianificazione della prevenzione o riguardanti la gestione delle emergenze; dall'altro lato, le blockchain utilizzate in simulazione permetterebbero la certificazione del dato della simulazione stessa e garantirebbero la fiducia reciproca tra le varie entità rappresentanti il comune, la provincia, i Vigili del Fuoco, ecc.

La prevenzione delle catastrofi naturali può essere condotta su due fronti:

1. il primo riguardante investimenti con la costruzione di strade, argini di contenimento di fiumi, eccetera;

2. il secondo riguardante una migliore regolazione della cementificazione in zone soggette a rischi idrogeologici, nivologici, sismici eccetera.

Per quanto riguarda il primo punto, l'utilizzo della blockchain permetterebbe ai vari enti di certificare nel database distribuito le azioni fatte per pianificare la prevenzione dei disastri naturali e, in caso di eventi quali alluvioni, terremoti, eccetera, certificare come tali misure si sono rivelate utili o insufficienti. In tal modo, avendo uno strumento di tracciamento di tutte le azioni preventive e della misura in cui si sono rivelate utili o insufficienti in caso di realizzazione dell'evento temuto, si potrebbe creare un più peciso modello di prevenzione e utilizzarlo per ottimizzare la pianificazione della prevenzione stessa.
Viceversa in una simulazione la certificazione del dato garantirebbe fiducia reciproca tra le entità coinvolte.

L'obiettivo delle riflessioni seguenti riguarda ora la gestione della cementificazione in zone soggette a disastri naturali (2.).

La motivazione nasce purtroppo da eventi disastrosi che avrebbero risparmiato vittime qualora si fosse evitata la costruzione di fabbricati in zone a rischio: un drammatico episodio è rappresentato dalla tragedia dell'Hotel Rigopiano, costruito ai piedi del Gran Sasso e spazzato via da una slavina provocando 29 vittime: l'albergo non avrebbe dovuto essere costruito in quel posto, come sostenuto, tra gli altri, dal Professor Gilberto Pambianchi, presidente nazionale dell'Associazione italiana di geomorfologia e professore ordinario dell'Università di Camerino [Secolo, 2017], e dall' Associazione H2o [Forum H2o, 2017], in quanto non solo costruito in una zona a rischio valanghe, ma per di più costruito sui detriti di valanghe precedenti.

Nel caso di Rigopiano sotto accusa anche la mancata realizzazione della Carta del pericolo delle valanghe.

Purtroppo anche in presenza di fatti oggettivi spesso per motivi economici la cementificazione non viene arrestata: solamente quattro giorni prima dell'alluvione del 25 ottobre 2011 nello Spezzino e nella Lunigiana (con conseguente inondazione delle zone della Val di Vara e della Val di Magra) fu votato in consiglio comunale il via libera alla costruzione dell'outlet di Brugnato, che

sarebbe dovuto sorgere proprio nella piana devastata dall'alluvione; una delibera della regione Liguria ha quindi bloccato per sei mesi le costruzioni nelle aree interessate, ma successivamente l'outlet è stato regolarmente costruito ed inaugurato a due anni e mezzo dalla tragedia.

Ciò che viene qui proposto è l'applicazione delle blockchain per regolare la cementificazione e ridurre la cementificazione illegale: la blockchain offre un sistema distribuito per la gestione della cementificazione; organizzazioni decentralizzate possono partecipare al processo decisionale riguardante il permesso o meno di costruire, operando nella rete senza bisogno di una terza parte.

L'idea è quella di utilizzare una consortium blockchain con protocollo PoV. In particolare, nel modello di seguito descritto, il consorzio è costituito dai comuni soggetti allo stesso tipo di rischio naturale, gli ordinary user sono i costruttori, i commissioner i periti comunali o provinciali, i butler gli esperti del settore (geologi, nivologi, sismologi, a seconda del tipo di criticità). Una transazione è rappresentata dalla richiesta a costruire di un costruttore-ordinary user, rivolta al perito locale-commissioner della propria zona; tale richiesta viene accettata dal perito locale-commissioner se egli la ritiene fat-

tibile e se la delibera comunale o provinciale è favorevole alla costruzione ed in tal caso, firmata dal commissioner, viene propagata nelle rete.
I nodi della rete ricevono le transazioni, ne verificano la validità e trasmettono quelle valide ai periti locali-commissioner e ai periti esperti-butler; i periti esperti-butler inseriscono le transazioni valide in una transaction pool. In ogni round di consenso il perito esperto-blutler nominato impacchetta in un blocco le transazioni della sua transaction pool che ritiene sensate dopo un'attenta analisi delle informazioni relative ad esse, e le invia ai periti locali-commissioner che lo firmano o lo respingono; se il blocco è firmato da più della metà dei commissioner il blocco è aggiunto alla blockchain e la costruzione viene autorizzata. Il numero di voti favorevoli ottenuti dai commissioner aumenta il punteggio del perito esperto-butler, punteggio che rappresenta il grado di fiducia da parte del commissioner.

Si potrebbe poi creare un modello agent based per simulare la regolazione della cementificazione mediante blockchain, ed uno senza blockchain per testare l'utilità della tecnologia. Nel primo caso si avrebbero 4 tipi di agenti (a rappresentare rispettivamente gli ordinary user, i butler, i butler candidate e i commissioner); tali agenti comunicherebbero attraverso la blockchain e nella blockchain inoltre verebbero definite le regole che permetteranno di

prendere la decisione.

Effettuando la simulazione si potrebbero confrontare i risultati per vedere se ad esempio, in casi di disastri realmente successi in passato, il modello proposto avrebbe respinto la costruzione, mentre il modello senza blockchain l'avrebbe permessa.

2.4 Esempio relativo al caso di studio: reclutamento di un perito per valutazione di un immobile oggetto di richiesta di mutuo

Come noto, per una banca la valutazione del valore di un immobile oggetto di richiesta di mutuo da parte di un perito è fondamentale: qualora il mutuatario non rispettasse le obbligazioni, se il mutuo erogato fosse molto superiore al valore di mercato dell'immobile, ciò comporterebbe alla banca una forte perdita.

In passato ogni banca si affidava ad uno stesso perito, per cui la probabilità di corruzione del perito stesso a favore di una valutazione maggiore dell'im-

mobile era talvolta alta.

Per ovviare il problema ora le banche chiedono al CRIF di nominare un perito per valutare gli immobili; non essendo noto prima, il problema precedente è ovviamente superato, ma rimane comunque il controllo centralizzato operato dal CRIF stesso.

Vogliamo ora vedere come l'applicazione del protocollo PoV ad una consortium blockchain potrebbe decentralizzare tale potere.

Nel modello proposto le filiali italiane di una stessa banca sono i nodi commissioner della rete blockchain utilizzante il protocollo PoV; i periti esperti sono i butler, mentre gli ordinary user sono i periti interni alle banche.

Una transazione consiste nella valutazione di un immobile da parte del perito locale-ordinary user; nel database essa sarà corredata dalle informazioni che hanno portato a tale stima e sarà firmata dalla relativa banca se ritenuta congrua. Tutte le transazioni saranno successivamente divulgate nella rete e ogni perito-butler inserirà quelle ritenute valide nella propria transaction pool. Tramite l'algoritmo di PoV un perito butler sarà scelto in maniera casuale per confezionare il prossimo blocco che, una volta creato, sarà votato dalle banche-commissioner che voteranno anche il perito-butler.

Conclusioni

Il lavoro svolto in questa tesi consta sostanzialmente di due parti.

Nella prima parte si è cercato di dare una descrizione della tecnologia blockchain indipendentemente dalle sue applicazioni finanziarie; sono stati poi analizzati gli algoritmi di consenso ad oggi noti, qui riuniti in una panoramica descrittiva e critica dello stato dell'arte su tale argomento, non nota al momento di questo studio. L'ordine espositivo di tali algoritmi è mosso dalla volontà di legare gli uni agli altri stabilendo confronti e sottolineando quali aspetti negativi dei precedenti abbiano motivato la nascita dei successivi. Nel descrivere gli algoritmi di consenso vengono messi in evidenza i punti di forza, le problematiche, la potenza avversaria tollerata, e viene effettuato un confronto tra i protocolli esaminati per quanto riguarda performance, scalabilità e resilienza agli attacchi.

Nella seconda parte si illustrano idee di applicazione della tecnologia block-

chain, in particolare per la previsione della domanda di una supply chain e per la prevenzione di disastri naturali.

L'applicazione della tecnologia blockchain per la previsione della domanda di una supply chain viene proposta su due fronti: da un lato nella *simulazione* di una supply chain, suggerendo l'utilizzo della tecnologia blockchain in un modello agent based di supply chain per garantire la certificazione del dato della simulazione e la fiducia reciproca tra le varie entità; dall'altro applicata ad una supply chain *reale*, per permettere l'identificazione e il tracciamento dei prodotti mediante informazioni convalidate e verificate, punto di partenza per la successiva creazione di un preciso modello di previsione della domanda e l'ottimizzazione della produzione e della logistica: viene quindi descritto un modello di supply chain basato su blockchain, descrivendo gli asset e le transazioni in questo contesto ed i ruoli dei vari nodi partecipanti alla rete blockchain.

Anche l'applicazione della tecnologia blockchain per la prevenzione delle catastrofi naturali viene proposta su due fronti: da un lato per permettere ai vari enti di certificare nel database distribuito le azioni fatte per pianificare la prevenzione dei disastri naturali e, qualora tali eventi si verificassero, certificare come tali misure si siano rivelate utili o insufficienti; dall'altro per

una migliore regolazione della cementificazione in zone soggette a rischi naturali: viene descritto l'utilizzo di una consortium blockchain con protocollo Proof of Vote con lo scopo di far partecipare organizzazioni decentralizzate al processo decisionale riguardante il permesso o meno di costruire, in modo da evitare la cementificazione in caso di rischio, nonostante la presenza di interessi economici.

L'idea di applicare una consortium blockchain per la prevenzione delle catastrofi naturali ha poi suggerito un'ulteriore idea di applicazione nel reclutamento di un perito per la valutazione di un immobile oggetto di richiesta di mutuo. In tal caso, essa permette di evitare valutazioni troppo superiori al valore di mercato dell'immobile che, qualora il mutuatario non rispettasse le obbligazioni, comporterebbero gravi perdite alle banche mutuanti.

Concludendo, se in primis il nostro lavoro fornisce un'analisi comparata degli algoritmi di consenso esistenti, utile a metterne in evidenza punti di forza e debolezza, esso illustra anche idee di applicazioni della tecnologia blockchain in vari campi (riguardanti supply chain, disastri naturali e mutui immobiliari), offrendo quindi suggerimenti interessanti per studi futuri atti a verificare tramite simulazione quanto detto.

Bibliografia

[Andrew, 2018] P. Andrew, What is Proof of Capacity? An Eco-Friendly Mining Solution, *https://coincentral.com/what-is-proof-of-capacity/* (2018).

[Andrew, 2018] P. Andrew, What is Burstcoin (BURST)? A complete guide, *https://coincentral.com/what-is-burstcoin-beginners-guide/* (2018).

[Antonopoulos, 2017] A. M. Antonopoulos, Mastering Bitcoin: Programming the Open Blockchain, *O' Really Media, Inc, 2nd edition* (2017).

[Armknecht et al., 2015] F. Armknecht, G.O. Karame, A. Mandal, F. Youssef, E. Zenner, Ripple: Overview and outlook, *Proc.*

Trust and Trusworthty computing (TRUST), vol. 9229 of Lecture Notes in Computer Science, Springer (2015), 163-180.

[Ateniese et al., 2013] G. Ateniese, I. Bonacina, A. Faonio, N. Galesi, Proofs of Space: When space is of the Essence, *https://eprint.iacr.org/2013/805/20140217:101935* (2013).

[Aublin et al., 2013] P. Aublin, S. B. Mokhtar, V. Quéma, RBFT: Redundant Byzantine Fault Tolerance, *2013 IEEE 33rd International Conference on Distributed Computing Systems (ICDCS)* (2013), 297-306.

[Back, 1997] A. Beck, Hashcash- A Denial of Service Counter-measure, *http://www.cypherspace.org/adam.* (1997).

[Ball et al., 2017] M. Ball, A. Rosen, M. Sabin and P. N. Vasudevan, Proofs of Useful Work, *https://eprint.iacr.org/2017/203.pdf.* (2017).

[Baran et al., 2008] I. Baran, E. D. Demaine, M. Patrascu, Subquadratic algorithms for 3SUM, *Algorithmica 50* (2008), 584-596.

[Barcelo, 2014] J. Barcelo, User Privacy in the Public Bitcoin Blockchain (2014).

[Benton et al., 2014] I. Bentov, C. Lee, A. Mizrahi, M. Rosenfeld, Proof of Activity: Extending Bitcoin's Proof of Work via Proof of Stake, *Proceedings of the ACM SIGmetrics 2014 Workshop on Economics of Networked Systems, NetEcon 2014.*

[Benton et al., 2016] I. Bentov, A. Gabizon, A. Mizrahi, Cryptocurrencies without Proof of Work, *Financial Cryptography and Data Security* Heidelberg: Springer (2016), 142-157.

[Bermingham, 2017] , Australian grain exporter completes successful blockchain pilots, *www.gtreview.com/news/asia/australian-grain-exporter-completes-successful-blockchain-pilots/* (2017).

[Bessani et al., 2014] A. N. Bessani, J. Sousa, E. A. P. Alchieri, State machine replication for the masses with BFT-SMaRt, *44th Annual IEEE/IFIP International Conference on Dependable Systems and Networks (DSN) 2014*(2014), 355-362.

[Biryukov et al., 2014] A. Biryukov, D. Khovratovich, I. Pustogarov, Deanonymisation of clients in Bitcoin p2p network, *Procee-*

dings of the 2014 ACM SIGSAC Conference on Computer and Communications Security, New York, NY, USA. (2014), 15-29.

[bitcoin forum, 2011] , Proof of stake instead of proof of work, *http://bitcointalk.org/index.php?topic=27787.0* (2011).

[Bowers et al., 2009] K.D. Bowers, A. Juels, and A. Oprea, Proof of of Retrievability: Theory and Implementation, *Proceedings of the 2009 ACM Workshop on on Cloud computing security. ACM, 2009* (2009), 43-54.

[Brewer, 2000] E. A. Brewer, Towards robust distributed systems, *ACM Symposium on Principles of Distributed Computing (PODC)* (2000), 4.

[Bruzzone et al., 2005] A. G. Bruzzone, R. Mosca, R. Revetria, E. Bocca, E. Briano, Agent Directed HLA Simulation for Complex Supply Chain Modeling. *SIMULATION: Transactions of The Society for Modeling and Simulation International, 81 (9).* (2015), 647-655.

[Bruzzone et al., 2018] A. G. Bruzzone, M. Massei, R. Di Matteo, K. Sinelshchikov M. Agresta, Models & Intelligent Agents

Interoperability via Blockchain based Simulations, *Technical Report of Simulation Team for Leonardo, Genoa, Italy* (2018).

[Butcher, 2018] M. Butcher, Verisart Brings Blockchain Certification to the Global Art Auction Market, *https://techcrunch.com/2018/05/03/verisart-brings-blockchain-certification-to-the-global-art-auction-market/* (2018).

[Buterin, 2015] V. Buterin, On Public and Private Blockchains, *Ethereum Blog.* (2015).

[Cachin et al, 2017] C. Cachin, M. Vukolić, Blockchain Consensus Protocol in the Wild, *https://arxiv.org/abs/1707.01873* (2017).

[Caldwell, 2002] C. Caldwell, Finding primes & proving primality, *https://primes.utm.edu/prove/merged.html* (2002).

[Castro et al., 2002] M. Castro, B. Liskov, Pratical Byzantine Fault Tolerance and Proactive Recovery, *ACM Transactions on Computer Systems, 20 (4)* (2002), 398-461.

[Chen et al., 2000] F. Chen, Z. Drezner, J.K. Ryan, D. Simchi-Levi, Quantifying the Bullwhip Effect in a Simple Supply Chain: The Impact of Forecasting, Lead Times, and Information, *Management Science 46 (4)* (2000), 436-443.

[Chen et al., 2000] F. Chen, Z. Drezner, J.K. Ryan, D. Simchi-Levi, The Impact of Exponential Smoothing Forecasts on the Bullwhip Effect, *Naval Research Logistics 47* (2000), 269-286.

[CoinDesk Inc, 2014] What is a Bitcoin Mining Pool?, *https: //www.coindesk.com/information/get-started-mining-pools* (2014).

[double spending, 2017] Irreversible Transactions, *https://en.bitcoin.it/wiki/Irreversible_transactions* (2017).

[Duan et al., 2014] S. Duan, H. Meling, S. Peisert, H. Zhang, BChain: Byzantine Replication with High Throughput and Embedded Reconfiguration, *Lecture Notes in Computer Science, vol 8878, Springer* (2014), 91-106.

[Dwork et al., 1988] C. Dwork, N. Lynch, L. Stockmeyer, Consensus in the Presence of Partial Synchrony, *Journal of the ACM, 32 (2)* (1988), 288-323.

[Dziembowski et al., 2013] S. Dziembowski, S. Faust, V. Kolmogorov and K. Pietrzak, Proof of Space, *in International Association for Cryptologic Research (IACR)* (2013).

[Ethereum] Ethereum Project, *https: //www.ethereum.org*

[Eyal et al., 2014] I. Eyal, E. G. Sirer, Majority is not Enough: Bitcoin Mining is Vulnerable, *Proceedings of International Conference on Financial Cryptography and Data Security, Berlin, Heidelberg* (2014), 436-454.

[Eyal et al., 2014] I. Eyal, E. G. Sirer, How to Disincentivize Large Bitcoin Mining Pools, *https: //hackingdistributed.com/2014/06/18/how-to-disincentivize-large-bitcoin-mining-pools* (2014)

[Eyal et al., 2016] I. Eyal, E. Gencer, E. G. Sirer, R. V. Renesse, Bitcoin-NG: A Scalable Blockchain Protocol, *Proceedings of*

the 13th Usenix Conference on Networked Systems Design and Implementation, Berkeley, CA (2016), 45-59.

[Forrester, 1961] J. W. Forrester, Industria Dynamics, *MIT Press* (1961).

[Forum H2o, 2017] Forum H2o, L'Hotel Rigopiano non doveva essere lì, è stato costruito sui resti di altre valanghe *https://www.abr24.it/forum-h2o-lhotel-rigopiano-non-doveva-essere-li-e-stato-costruito-sui-resti-di-altre-valanghe/* (2017).

[Galvin, 2017] D. Galvin, IBM and Walmart: Blockchain for Food Safety, *PowerPoint presentation* (2017).

[Gao et al., 2016] J. Gao, R. Impagliazzo, Orthogonal Vectors is Hard for

First-Order Properties on Sparse Graphs, *Electronic Colloquium on Computational Complexity (ECCC), vol. 23* (2016), 53.

[Ghost, 2015] A. Zolar, Y. Sompolinsky, Secure High-Rate Transaction Processing in Bitcoin, *Financial Chryptography and Data Security.*

FC 2015., Lecture Notes in Computer Science, vol 8975. Springer, Berlin, Heidelberg (2015).

[Hearn, 2016] M. Hearn, Corda: A Distributed ledger, *http://www.corda.net/content/corda-technical-whitepaper.pdf* (2016).

[Hern, 2018] A. Hern, Bitcoin's energy usage is huge-we can't afford to ignore it, *https://www.theguardian.com/technology/2018/jan/17/bitcoin-electricity-usage-huge-cliate-cryptocurrency* (2018)

[Holland, 1992] J.H. Holland, Adaptation in Natural and Artificial Systems, *MIT Press* (1992).

[Hyperledger] Hyperledger project, *https: //www.hyperledger.org* (2015)

[Hyperledger] Hyperledger Architecture, Volume 1, *https://www.hyperledger.org/wp-content/uploads/2017/08/Hyperledger_Arch_WG* (2017)

[Johnson et al., 2001] D. Johnson, A. Menezes, S. Vanstone, The Elliptic Curve Digital Signature Algorithm (ECDSA), *International Journal of Information Security 1 (1)*(2001), 36-63.

[Juels et al., 2007] A. Juels, B. S. Kaliski, PORs: Proofs of Retrievability for Larges Files, *Proceedings of the 14th ACM conference on Computer and communications security, Alexandria, Virginia, USA.* (2007).

[Kiayias et al., 2017] A. Kiayias, A. Russel, B. David, R. Oliynykov, Ouroboros: A Provably Secure Proof-of-Stake Blockchain Protocol, *Advances in Cryptology-CRYPTO 2017.* Cham, Switzerland: Springer (2017), 357-388.

[Kim et al, 2018] H. M. Kim, M. Laskowski, Toward an Ontology Driven Blockchain Design for Supply Chain Provenance, *ACM Intelligent Systems in Accounting, Finance and Management, 25(1)*, (2018), 18-27.

[Kimbrough et al, 2002] S.O. Kimbrough, D.J. Wu, F. Zhong, Computers Play the Beer Game: Can Artificial Agents Manage Supply Chains?, *Decision Support Systems 33*, (2002), 323-333.

[King et al, 2012] S. King, S. Nadal, PPcoin: Peer-to-Peer Crypto-Currency with Proof-of-Stake, *https: //www.decred.org/research/king2012.pdf.* (2012)

[King, 2013] S. King, Primecoin: Cryptocurrency with Prime Number Proof-of-Work, *https: //www.primecoin.io/bin/primecoin-paper.pdf* (2013)

[Kosba et al., 2016] A. Kosba, A. Miller, E. Shi, Z. Wen, C. Papamanthou, Hawk: The Blockchain Model of Cryptography and Privacy-Preserving smart contracts, *Proceedings of IEEE Symposium on Security and Privacy, San Jose, CA, USA.* (2016), 839-858.

[Krajewski et al, 2002] L.J. Krajewski, L.P. Ritzman, Operations Management: Strategy and Analysis, *Prentice Hall* (2002).

[Kreku et al., 2017] J. Kreku, V. Vallivaara, K. Halunen, J. Suomalainen, Hawk: Evaluating the Efficiency of Blockchains in IoT with Simulations, *2nd International Conference on Internet of Things, Big Data and Security, IoTBDS 2017.* (2017), 216-223.

[Lamport et al., 1982] L. Lamport, R. Shostak, M. Pease, The Byzantine Generals Problem, *ACM Transactions on Programming Languages and Sistems, 4 (3)* (1982), 382-401.

[Larimer, 2014] D. Larimer, Delegated Proof-of-Stake (DPOS) , *http://bitshares.org/technology/delegated-proof-of-stake-consensus/* (2014)

[Lee et al., 1993] H. L. Lee, C. Billington, Material Management in Decentralized Supply Chains, *Operations Research 41 (5)* (1993), 835-847.

[Lee et al., 1997] H. L. Lee, V. Padmanabhan, S. Whang, The Bullwhip Effect in Supply Chains, *Sloan Management Review 38 (3)* (1997), 93- 102.

[Lee et al., 1997] H. L. Lee, V. Padmanabhan, S. Whang, Information Distortion in a Supply Chain: The Bullwhip Effect, *Management Science 43 (4)* (1997), 546.

[Lee et al., 1999] H.L. Lee, S. Whang, Decentralized Multi-Echelon Supply Chains: Incentives and Information, *Management Science 45 (5)* (1999), 663- 640.

[Li et al., 2017] K. Li, H. Li, H. Hou, K. Li, Y. Chen, Proof of Vote: A High Performance Consensus Protocol Based on Vote Mechanism & Consortium Blockchain, *IEEE 19nd International Conference on High Performance Computing and Communications* (2017), 465,473.

[Liang et al., 2006] W.Y. Liang, C. C. Huang, Agent-based demand forecast in multi-echelon supply, *Decision Support System 42* (2006), 390- 407.

[Liang et al, 2017] X. Liang, S. Shetty, D. Tosh, C. Kamhoua, K. Kwiat, L. Njilla, ProvChain: A Blockchain-Based Data Provenance Architecture in Cloud Environment with Enhanced Privacy and Availability, *Proceedings of the 17th IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing. IEEE Press.*, (2017), 468-477.

[Lifchitz, 1998] H. Lifchitz, Generalization of Euler-Lagrange theorem, *http://www.primenumbers.net/Henri/us/NouvTh1us.htm* (1998)

[Liu et al, 2017] Z. Liu, S. Tang, S. M. Chow, Z. Liu, Y. Long, Forking-Free Hybrid Consensus with Flexible Proof-of-Activity,

http://eprint.iacr.org/2017/367.pdf (2017)

[Liu et al, 2017] B. Liu, X. L. Yu, S. Chen, X. Xu, L. Zhu . Blockchain Based Data Integrity Service Framework for IoT Data. *2017 IEEE International Conference on Web Services (ICWS) Honolulu, Hawaii, USA*(2017),468-475.

[McBurney et al., 2002] P. McBurney, S. Parsons, J. Green, Forecasting market demand for new telecommunications services: an introduction, *Telematics and Informatics 19* (2002), 225-249.

[Meiklejohn et al., 2013] S. Meiklejohnl, M. Pomarole, G. Jordan, K. Levchenko, D. McCoy, G. M. Voelker, S. Savage, A Fistful of Bitcoins: Characterizing Payments Among Men with No Names, *Proceedings of the 2013 Conference on Internet Measurement Conference, New York.* (2013).

[Merkel, 1979] R. C. Merkel, Secrecy, Authentication and Public Key Systems, *PhD Thesis, Dept. of Electrical Engineering, Stanford University.* (1979).

[Michalewicz, 1996] Z. Michalewicz, Genetic Algorithm+Data Structures=Evolution Program 3rd, *Springer, New york.* (1996).

[Miller et al., 2015] A. Miller, A. Kosba, J. Katz, E. Shi, Nonoutsourceable Scratch-Off Puzzles to Discourage Bitcoin Mining Coalitions, *Proceedings of the 22nd ACM SIGSAC Conference on Computer and Communications Security , New York, NY.* (2015), 680-691.

[Milutivonic et al., 2016] M. Milutivonic, W. He, H. Wu, M. Kanwa, Proof of Luck: an Efficient Blockchain Consensus Protocol *https://eprint.iacr.org/2017/249.pdf.* (2016)

[Ministero della Salute, 2015] Ministero della Salute, Manuale di Corretta Prassi Igienica delle Aziende che Operano nello Stoccaggio e Distribuzione di Prodotti Alimentari, *http://www.salute.gov.it/imgs/C_17_pubblicazioni_2521_allegato.pdf.*

[motherboard] bitcoin coulds consume as much electricity as denmark by 2020, *https://motherboard.vice.com/en_us/article/aek3za/bitcoin-could-consume-as-much-electricity-as-denmark-by-2020.*

[Nakamoto, 2008] S. Nakamoto, Blockchain: Bitcoin: A peer-to-peer Electronic Cash System *O' Really Media, Inc.* (2008).

[New et al., 1995] S. J. New, P. Payne, Research frameworks in logistics: three models, seven dinners and a survey, *International Journal*

of Phisical Distribution and Logistics Management 25 (10) (1995), 60-67.

[Nxt] Nxt, *https://nxtplatform.org*

[P4Titan, 2014] P4Titan, Slimcoin A Peer-to-Peer Crypto-Currency with Proof-of-Burn Mining without Powerful Hardware , *https://www.chainwhy.com/upload/default/20180703/4ae7cee40462e7951f508b28dd1d9936.pdf* (2014).

[Park et al., 2015] S. Park, A. Kwon, J. Alwen,G. Fuchsbauer, P. Gazi, K. Pietrzak, Spacemint: A Cryptocurrency Based on Proofs of Space, *https://eprint.iacr.org/2015/528.pdf* (2015).

[Pass et al., 2017] R. Pass, E. Shi, Hybrid Consensus: Efficient Consensus in the Permissionless Model, *IACR Cryptology ePrint 2016/917* (2017).

[Pawlak, 1991] Z. Pawlak, Rough Sets, *Kluwer Academic Publishers, Dordrecht, The Netherlands,* (1991).

[Ramachandran et al., 2017] A. Ramachandran, D. Kantarcioglu, Using Blockchain and smart contracts for secure data provenance management, *https://arxiv.org/abs/1709.10000* (2017).

[Ren, 2014] L. Ren, Proof of Stake Velocity: Building the Social Currency of the Digital Age, *https://www.reddcoin.com/papers/PoSV.pdf* (2014).

[Rilee, 2018] K. Rilee, Understanding Hyperledger Sawtooth - Proof of Elapsed Time, *https://medium.com/kokster/understanding-hyperledger-sawtooth-proof-of-elapsed-time-e0c303577ec1* (2018).

[Sapirsthein et al , 2015] A. Sapirsthein, Y. Sompolinsky, A. Zohar, Optimal Selfish Mining Strategies in Bitcoin. *CoRR, abs/1507.06183* (2015).

[Schwartz et al., 2017] D. Schwartz, N. Youngs, A. Britto, The Ripple Protocol Consensus Algorithm, *https://www.ripple.com/files/ripple_consensus_whitepaper.pdf* (2014).

[Scott et al., 1991] C. Scott, R. Westbrook, New Strategic Tools for Supply Chain Management, *International Journal of Phisical Distribution and Logistics Management 21 (1)* (1991), 23-33.

[Seang et al,2018] S. Seang, D. Torre, Proof of Work and Proof of Stake consensus protocol: a blockchain application for local complementary currencies , *https://gdre-scpo-aix.sciencesconf.org/195470/document* (2018).

[Secolo, 2017] Secolo d'Italia, La denuncia di un esperto: quell' Hotel non doveva essere costruito lì... , *http://www.secoloditalia.it/2017/01/431327/* (2017).

[Seidel, 1992] R. Seidel,On the all-pairs-shortest-path problem, *Proceeding STOC '92 Proceedings of the twenty-fourth annual ACM symposium on Theory of computing* (1992), 745-749.

[SHA256, 2016] SHA-256, *https://en.bitcoin.it/wiki/SHA-256* (2016).

[Shacham et al., 2008] H. Shacham, B. Waters, Compact Proofs of Retrievability, *Proceedings of the 14th conference on the Theory and Application of Cryptology and Information Security: Advances in Cryptology, Melbourne, Australia.* (2008).

[Shoker, 2017] A. Shoker, Sustainable Blockchain through Proof of eXercise, *in Network Computing and Applications (NCA), 2017 IEEE 16th International Symposium, Cambridge.* (2017).

[Simchi-Levi et al., 2000] D. Simchi-Levi, P. Kaminsky, E. Simchi-Levi, Designing and Managing the Supply Chain Concepts, Strategies, and Case Studies, *McGraw-Hill, Boston*, (2000).

[Sompolinsky et al, 2015] Y. Sompolinsky, A. Zohar, Secure High-Rate Transaction Processing in Bitcoin, *Financial Cryptography and data security-19th International Conference, FC 2015.* (2015), 507-527.

[Sterman, 1988] J. D. Sterman, Modeling Managerial Behaviour: Misperceptions of Feedback in a Dynamic Decision Making Experiment, *Management Science, 35 (3)* (1988), 321-339.

[Sterman et al, 1992] J. D. Sterman, E. Mosekilde, J. S. Thomsen, Hyperchaotic Phenomena in Dynamic Decision Making, *S AMS, 9* (1992), 137-156.

[Stewart, 2012] I. Stewart, Proof of burn - a potential alternative to proof of work and proof of stake *https://bitcointalk.org/index.php?topic=131139.0*(2012)

[Swan, 2015] M. Swan, Blockchain: Blueprint for a New Economy *O' Really Media, Inc.* (2015).

[Swanson, 2015] T. Swanson, Consensus-as-a-service: a brief report on the emergence of permissioned, distributed ledger systems. *https://www.ofnumbers.com/2015/04/06/consensus-as-a-service-a-brief-report-on-the-emergence-of-permissioned-distributed-ledger-systems/*(2015)

[Tendermint] Tendermint,*https://tendermint.com/*.

[Tromp, 2014] J. Tromp, Cuckoo cycle: a memory-hard proof-of-work system, *https://eprint.iacr.org/2014/059.pdf* (2014).

[Tromp] J. Tromp, Cuckoo Hashing, *https://web.stanford.edu/class/cs166/lectures/13/Small13.pdf*.

[Umeda et al, 1998] S. Umeda, A. Jones . An Integration Test-bed System for Supply Chain Management, *Proceedings of the 1998 Winter Simulation Conference, 1998*, D.J. Medeiros, E.E. Watson, J.S. Carson, M.S. Manivannan Eds. (1998), 1337-1385.

[Vasin, 2014] P. Vasin, BlackCoin's Proof-of-Stake Protocol v2, *https://blackcoin.co/blackcoin-pos-protocol-v2-whitepaper.pdf* (2014).

[Yasaweerasinghelage et al, 2017] R. Yasaweerasinghelage, M. Staples, I. Weber . Predicting Latency of Blockchain-Based Systems Using Architectural Modelling and Simulation, *2017 IEEE International Conference on Software Architecture (ICSA) Gothenburg, Sweden* (2017),253-256.

[Zhao et al., 1993] X. Zhao, T.S. Lee, Freezing the master production schedule in multilevel material requirements planning systems under demand uncertainty, *Journal of Operations Management 11* (1993), 185-2015.