Jollar*
una criptovaluta in Jolie

Stefano Pio Zingaro

May 7, 2018

Abstract

Il progetto di quest'anno propone la creazione un sistema di scambio elettronico decentralizzato, dove gli utenti effettuano transazioni certificate all'interno della rete stessa, senza il bisogno di un organo centrale garante[1]. L'implementazione del progetto segue i principi della programmazione orientata ai servizi, rispettandone l'architettura e i paradigmi di gestione della concorrenza. La comunicazione tra i servizi, infine, è implementata nel linguaggio visto a lezione di laboratorio: Jolie.

Indice

*Jollar è un gioco di parole che viene da Jolie + Dollar.

[1]Il ruolo che ricoprono le banche nei moderni sistemi finanziari.

1 Informazioni Logistiche

In questa sezione sono riportate le informazioni logistiche a riguardo del progetto: un *reminder* sulla formazione dei gruppi; un elenco di possibili date per la consegna dell'implementazione e del report. Tali informazioni sono soggette a cambiamenti ed a revisioni, ogni modifica viene comunicata attraverso la pagina ufficiale del corso, la pagina del tutor, ed il newsgroup. È possibile chiedere delucidazioni e prenotare un ricevimento via messaggio di posta elettronica, specificandone la motivazione, a questo indirizzo.

1.1 Formazione dei Gruppi

I gruppi sono costituiti da un minimo di tre ad un massimo di cinque persone, coloro che intendono partecipare all'esame comunicano entro e non oltre il **10 Maggio 2018** (pena esclusione dall'esame) la composizione del gruppo di lavoro, **via posta elettronica**, all'indirizzo `stefanopio.zingaro@unibo.it`. Il messaggio ha come oggetto **GRUPPO LSO** e contiene:

1. Il nome del gruppo;
2. Una riga per ogni componente: cognome, nome e matricola;
3. Un indirizzo di posta elettronica di riferimento a cui mandare le notifiche, è incarico del proprietario trasmetterle agli altri membri.

Email di esempio con oggetto **GRUPPO LSO**

NomeGruppo

- *Vader Darth, 123456*
- *Pallino Pinco, 234567*
- *Banana Joe, 345678*

Referente: joe.banana@studio.unibo.it

Chi non riuscisse a trovare un gruppo invia allo stesso indirizzo di posta elettronica un messaggio con oggetto **CERCO GRUPPO LSO**, specificando:

1. Cognome, Nome, Matricola, Email;
2. Eventuali preferenze legate a luogo e tempi di lavoro (si cercherà di costituire gruppi di persone con luoghi e tempi di lavoro compatibili)

Email di esempio con oggetto **CERCO GRUPPO LSO**

Pallino Pinco, 234567, preferirei nei pressi del dipartimento, tutti i giorni dopo pranzo.

Le persone senza un gruppo vengono assegnate il prima possibile senza possibilità di ulteriori modifiche.

1.2 Date di consegna e dell'Orale

Ci sono due date disponibili per la consegna dell'implementazione e del report. Queste sono:

- le 23.59 di **Lunedí 2 Luglio** 2018;
- le 23.59 di **Lunedí 17 Settembre** 2018.

La data presa in considerazione per la consegna della parte di implementazione sarà quella di creazione del **Tag** su GitLab (le istruzioni più avanti nel testo). Solo in seguito alle consegne, vengono fissate data ed orario della discussione (notificate tramite la mail di riferimento), compatibilmente coi tempi di correzione. La discussione dell'implementazione e la relativa demo di funzionamento viene effettuata in un incontro unico con tutti i componenti presenti. Al termine della discussione, ad ogni singolo componente verrà assegnato un voto in base all'effettivo contributo dimostrato nel lavoro. La valutazione è indipendente dal numero di persone che compongono il gruppo.

2 Componenti del Progetto e loro Implementazione

In generale, il sistema consiste in una rete *Peer-To-Peer* (**P2P**) che utilizza *Proof-of-Work* (**POW**) per registrare l'elenco delle transazioni in un archivio pubblico, detto **blockchain**. Per una migliore comprensione del fenomeno si rimanda alla lettura dell'articolo originale di Satoshi Nakamoto Bitcoin. L'implementazione del progetto è, almeno in parte, ripresa da quella descritta nell'articolo, fatta eccezione per lo sviluppodella Proof-of-Work, che segue i principi della criptovaluta **Primecoin**.

In questa sezione sono enunciate le definizioni di transazioni (contenute in un blocco) e blockchain, e se ne discute la relativa implementazione. Sono poi elencate le definizioni delle altre componenti del sistema: il meccanismo di convalida di un blocco (la Proof-of-Work); il server Timestamp per la sincronizzazione oraria; il comportamento della rete all'avvio dell'esecuzione.

2.1 Le transazioni e la catena di blocchi: la Blockchain

La **blockchain** è una struttura dati ordinata, una lista concatenata di blocchi contenenti transazioni. L'intera struttura della blockchain può essere conservata in un file, oppure in un database. Un **block**, o blocco, è un contenitore, una struttura dati a sua volta, che aggrega transazioni che devono essere incluse in un *ledger*[2] pubblico e condiviso, la blockchain. Il blocco è composto da un *header*, che contiene i *metadata*, e dalla lista di transazioni che ne compongono la maggior parte della grandezza. Di seguito un esempio in Jolie di un tipo "custom".

2.1.1 Implementazione

1. L'ID del blocco;
2. Il hash del blocco precedente;
3. Lunghezza della Proof-of-Work;
4. Il numero primo origine (descritto più avanti);
5. La catena di numeri primi che compone la Proof-of-Work di quel blocco;

[2]Il termine ledger indica il libro mastro dei contabili, viene tradotto in italiano come registro.

```
type Blockchain: void {
  .block*: Block
}

type Block: void {
  .previousBlockHash: string
  .difficulty: double
  .transaction*: Transaction
}

type Transaction: void {
  .nodeSeller: Node
  .nodeBuyer: Node
  .jollar: int
}

type Node: void {
  .publicKey: string
  .privateKey?: string
}
```

In Jolie la funzione di Hash MD5 (da 128 bit) si trova già implementata nell'interfaccia `MessageDigest`.

```
include "message_digest.iol"
include "console.iol"

main
{

  //...
  md5@MessageDigest( "secret" )( response )
  ;
  println@Console( response )()
  //...
}
```

2.2 Il Server Timestamp

Un server timestamp offre un servizio che permette di conoscere l'ora esatta all'interno della rete. Un nodo che intende effettuare le operazioni di scrittura o di validazione di un blocco deve richiedere il **timestamp** al server. Esso garantisce l'esistenza dei dati al momento della richiesta.

2.2.1 Implementazione

In particolare, Jolie offre, tramite l'interfaccia `time` una `operation` che restituisce i millisecondi del *current unix timestamp* nel momento esatto della richiesta. Tramite una chiamata a tale servizio si può ottenere il dato necessario da includere nel blocco. Qui è riportato un esempio si chiamata a tale operation, che può essere inserita in un server che accetta richieste e le inoltra al servizio time.

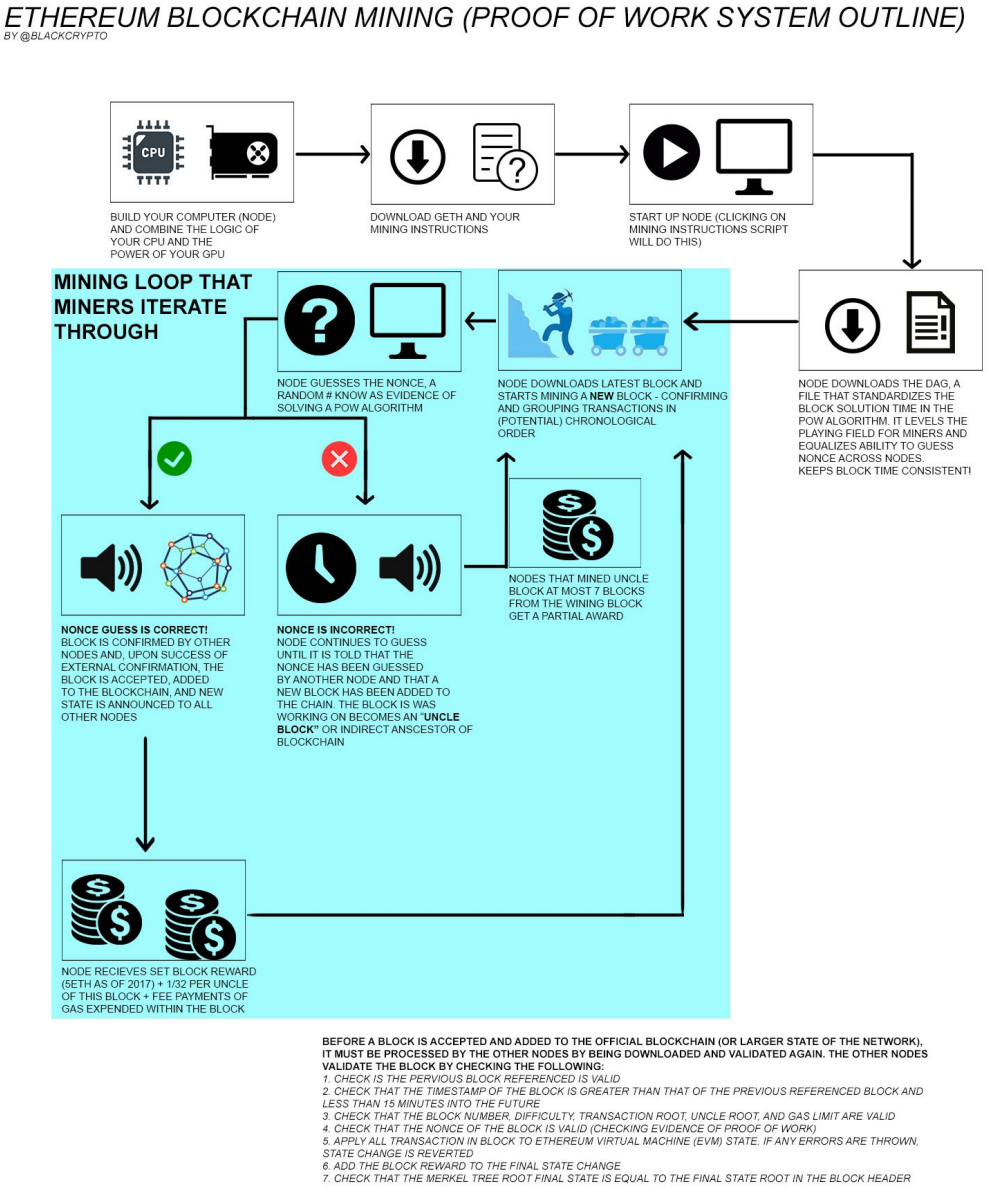
```
include "time.iol"

main
{
  //...
  getCurrentTimeMillis@Time()( millis )
  ;
  block.timestamp = millis
  //...
}
```

2.3 La Proof-of-Work

La Proof-of-Work, è un algoritmo che viene utilizzato per raggiungere un accordo decentralizzato tra diversi nodi nel processo di aggiunta di un blocco specifico alla blockchain. Tale algoritmo produce valori che vengono utilizzati per verificare che sia stata eseguita una notevole quantità di lavoro. Nell'immagine riportata qui sotto viene visualizzato un esempio di utilizzo di tecniche Proof-of-Work nel mondo delle criptovalute (in particolare quella di Ethereum ma generalizzabile a qualunque altra che usa Proof-of-Work). Per implementare un sistema decentralizzato basato su rete P2P, abbiamo bisogno di usare un sistema di Proof-of-Work[3]. Questo metodo consiste nell'obbligare i nodi che vogliono scrivere un blocco a cercare un valore che sia difficile da trovare e di cui sia facile controllarne la correttezza da parte degli altri nodi che vogliono validare la scrittura.

[3] http://www.wisdom.weizmann.ac.il/~naor/PAPERS/pvp.ps



2.3.1 Implementazione

La Proof-of-Work scelta per produrre blocchi ed ottenere così sei Jollar in *reward*[4] è quella utilizzata dalla moneta PrimeCoin. Essa consiste nel gener-

[4] A differenza di Bitcoin, il reward in questo caso è fissato a sei ed è costante per tutta la vita della rete, un *improvement* possibile è cambiare questo valore in una funzione decrescente.

are delle catena di numeri primi con determinate proprietà. Solo una catena che appartiene ad una di queste tre categorie può essere validata, la POW consiste quindi nel creare una sola di queste catene, il nodo avrà quindi tre possibilità.

Catene di Cunningham del Primo Tipo

Una catena di Cunningham del primo tipo di lunghezza n è un array di numeri primi:

$$(p_1, \ldots, p_n) \text{ tale che per ogni } 1 \leq i < n \text{ allora } p_i + 1 = 2p_i + 1$$

Il numero p_1 è chiamato l'**origine** della catena. L'origine di una catena va inclusa nel blocco! Segue dalla regola:

$$\begin{aligned} p_2 &= 2p_1 + 1, \\ p_3 &= 4p_1 + 3, \\ p_4 &= 8p_1 + 7, \\ &\ldots \\ p_i &= 2^{i-1}p_i + (2^{i-1} + 1) \end{aligned} \tag{1}$$

Un esempio pratico con $p_1 = 2$ e lunghezza $n = 5$:

$$\begin{aligned} p_1 &= 2 \\ p_2 &= 2p_1 + 1 = 2 * 2 + 1 = 5, \\ p_3 &= 4p_1 + 3 = 4 * 2 + 3 = 11, \\ p_4 &= 8p_1 + 7 = 8 * 2 + 7 = 23, \\ &\ldots \\ p_i &= 2^{i-1}p_i + (2^{i-1} + 1) = 2^4 * 2 + (2^4 - 1) = 47 \end{aligned} \tag{2}$$

Catene di Cunningham del Secondo Tipo

Una catena di Cunningham del secondo tipo di lunghezza n è un array di numeri primi:

$$(p_1, \ldots, p_n) \text{ tale che per ogni } 1 \leq i < n \text{ allora } p_i + 1 = 2p_i - 1$$

Il numero p_1 è chiamato l'**origine** della catena. L'origine di una catena va inclusa nel blocco! Segue dalla regola:

$$\begin{aligned} p_2 &= 2p_1 - 1, \\ p_3 &= 4p_1 - 3, \\ p_4 &= 8p_1 - 7, \\ &\dots \\ p_i &= 2^{i-1}p_i - (2^{i-1} - 1) \end{aligned} \tag{3}$$

Catene Bi-twin

Una catena di Bi-twin di lunghezza $k+1$ è un array di numeri primi:

$$n-1, n+12n-1, 2n+1, \dots, 2^k n-1, 2^k n+1$$

Un esempio:

- origine: $n = 5$
- lunghezza: $k = 5$
- catena: $5, 7, 11, 13$

Le catene Bi-twin posso anche essere viste come l'unione di due catene di Cunningham del primo e del secondo tipo con la stessa origine e lunghezza. In questo caso viene scelta l'origine della prima catena essendo la minore.

Validazione delle catene

Il meccanismo di controllo e validazione delle Proof-of-Work inviate contenenti le catene di numeri primi prevede un test che utilizza il piccolo Teorema di Fermat (link alla dimostrazione), esso afferma che, per p primo ed a intero

$$a^{p-1} \equiv 1 (\mathrm{mod} p) \text{ se } p \text{ non divide } a.$$

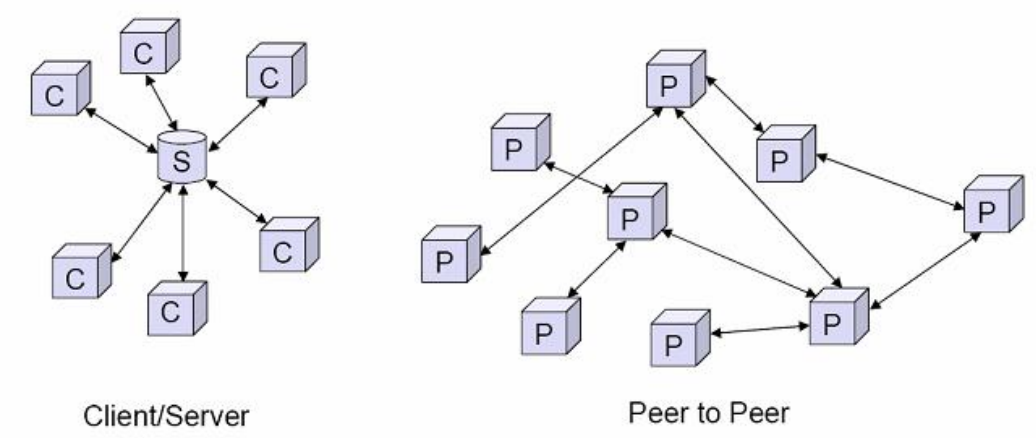
I p che soddisfano questo proprietà vengono chiamati *pseudoprimi*, perché non tutti sono, appunto, primi. Vale però che maggiore è p, maggiore è la probabilità che esso sia primo.
Un esempio utilizza tipicamente $a = 2$ e lo eleva alla $p - 1$ (prendiamo 17 e 23) che modulo p risulta essere uguale ad 1:

$$2^{16} = 65536 \bmod 17 = 1$$
oppure
$$2^{22} = 4194304 \bmod 23 = 1$$

Ciò significa che per la primalità va controllata per ogni numero primo della catena. Inoltre la validazione controlla anche che la lunghezza della catena sia maggiore o uguale alla difficoltà (quindi il tipo di catena va incluso nel blocco!)

2.4 La Rete Peer-To-Peer

Una rete P2P, come ad esempio quella di Emule, è una rete i cui nodi non sono **client** o **server** fissi, ma prendono forma di entità equivalenti o "paritarie".



2.4.1 Implementazione

L'esecuzione della rete P2P comprende i seguenti passaggi:

1. Le nuove transazioni sono inviate in *broadcast*[5] a tutti i nodi che partecipano alla rete;
2. Ogni nodo colleziona la nuove transazione in un blocco (una transazione per blocco!);
3. Ogni nodo effettua del "lavoro" per provare di poter scrivere il blocco (Proof-of-Work);
4. Quando un nodo trova una prova, invia il blocco in broadcast a tutti i nodi;

[5]Nelle reti di calcolatori, il termine broadcast indica una modalità di instradamento per la quale un pacchetto dati inviato ad un indirizzo particolare (detto appunto di broadcast) verrà consegnato a tutti i computer collegati alla rete

5. I nodi accettano il nuovo blocco solo se la transazione contenuta in esso è valida e non è stata già "spesa";

I nodi considerano sempre la blockchain più lunga come quella corretta. Nel caso in cui due nodi inviano versioni differenti del nuovo blocco contemporaneamente, alcuni nodi potrebbero riceverne una versione, altri l'altra. In quel caso il nodo lavora sul blocco ricevuto temporalmente prima. Il nodo che ha lavorato su un blocco non sincronizzato se ne rende conto quando una nuova validazione viene richiesta, in questo caso egli manda in broadcast una richiesta di sincronizzazione sul blocco in questione e, una volta ricevuta risposta, aggiorna la blockchain alla versione più recente, potendo così validare il nuovo blocco ricevuto.

2.5 Il Network Visualiser

Il Network Visualiser è un tool amministrativo da terminale per il monitoraggio del sistema.

2.5.1 Implementazione

Il Network Visualizer invia richieste a tutta la rete per conoscere lo stato delle blockchain di tutti i nodi, la loro versione, le transazioni che contengono, generando le seguenti informazioni:

- La data e l'ora del server timestamp al momento della richiesta.
- Per ogni utente/nodo: la sua chiave pubblica; le transazioni che ha effettuato divise per entrate/uscite; la versione della blockchain che utilizza e la quantità totale di Jollar che possiede.
- Per ogni transazione al punto precedente, ordinate in ordine cronologico, si indica la data, l'ora, e gli utenti (con le loro chiavi pubbliche) coinvolti.
- La blockchain più lunga presente nella rete (chiamata la versione ufficiale), con tutta la struttura dati che contiene.

3 Specifiche di Consegna

Globalmente vengono consegnati tre prodotti: la documentazione del progetto, la sua implementazione in codice Jolie ed una demo di funzionamento. La valutazione finale avviene mediante una discussione di gruppo.

N.B.

1. non si accettano richieste di eccezioni sui progetti con motivazioni legate a esigenze di laurearsi o di non voler pagare le tasse per un altro anno.
2. chi copia o fa copiare, anche solo in parte, si vede invalidare completamente il progetto senza possibilità di appello → deve quindi rifare un nuovo progetto l'anno successivo.

3.1 La documentazione

È possibile scrivere la documentazione nel formato preferito, l'importante è che il PDF generato rispetti la struttura del modello (riportato qui in basso). La documentazione ha lunghezza di quattro o cinque pagine (quindi da 8 a 10 facciate), è scritto con font di grandezza **12pt** e viene consegnato in formato PDF. Di seguito viene riportato un esempio di documentazione con le principali caratteristiche da inserire.

L'intestazione della Documentazione

- Jollar – Laboratorio Sistemi Operativi A.A. 2017-2018
- Nome del Gruppo
- Indirizzo mail di riferimento: myaccount@email.com
- Componenti:
  - Cognome, Nome, Matr. 0000424242
  - ...

Il corpo della Documentazione

1. Descrizione generale del progetto – descrizione delle features implementate e del contenuto della documentazione.
2. Istruzioni per la demo – le istruzioni per eseguire una demo.
3. Discussione sulle strategie di implementazione
   (a) Struttura del progetto – come è stato diviso il progetto, perché, i problemi principali riscontrati, le alternative considerate e le soluzioni scelte.
   (b) Sezione di descrizione della feature x – abbiamo implementato la funzione di 'foo' ... (con esempi di codice).

3.1.1 Griglia di Valutazione

La valutazione della documentazione verte sull'analisi dello scritto e sulla sua capacità di esprimere con chiarezza i concetti descritti, soprattutto grazie all'uso di esempi. In particolare la griglia di valutazione usata è la seguente:

Qualità dell'informazione	Riconoscimento dei problemi (di concorrenza) e loro descrizione.
Uso degli esempi	Presenza di almeno un esempio in tutte le scelte implementative.
Analisi delle scelte implementative	Descrizione della propria scelta implementativa e presenza di proposte di alternative valide.

3.2 L'Implementazione

Il progetto viene sviluppato utilizzando il linguaggio Jolie. Non ci sono requisiti riguardo ai protocolli (*protocol*) e i media (*location*) utilizzati per realizzare la comunicazione tra i componenti del sistema. La gestione del progetto avviene col supporto del sistema *git*. Il codice del progetto è contenuto in un repository su GitLab e viene gestito seguendo la procedura descritta sotto.

GitLab

- Ogni membro del gruppo crea un account su GitLab;
- il referente del gruppo crea un nuovo progetto cliccando sul + in alto a destra nella schermata principale di GitLab, inserendo il nome "LabSO_NomeGruppo" e cliccando su **New Project**;
- una volta che il progetto è stato creato, il referente aggiunge ogni membro del gruppo come **role permission** >**Developer** al progetto andando su **Settings** > **Members** nel menù a sinistra, cercandoli in base allo username registrato su GitLab;
- il referente aggiunge l'utente "stefanopiozingaro" come **role permission** > **Reporter**.

Al momento della consegna, il repository dovrà contenere i sorgenti del progetto e la relazione, nominata **REPORT_LSO.pdf**. Per effettuare la consegna:

1. nella pagina di GitLab del repository, cliccare sulle voci del menù **Repository** > **Tags** > **New Tag**;
2. digitare come **Tag Name** il nome **Consegna**;
3. cliccare su **Create Tag** per eseguire la creazione del **Tag** di consegna.

Una volta creato il Tag, inviare una email di notifica di consegna con soggetto **CONSEGNA LSO - NOME GRUPPO** a questo indirizzo di posta elettronica.

N.B.

La documentazione va consegnata in forma cartacea nella casella del prof. Sangiorgi al piano terra del Dipartimento di Informatica, a fianco del suo ufficio.

3.2.1 Griglia di Valutazione

La valutazione dell'implementazione del sistema si basa sull'analisi del codice Jolie, sull'uso dei costrutti del linguaggio per la creazione di soluzioni efficienti, sulla tolleranza ai guasti del sistema implementato e sulla gestione degli errori. In particolare la griglia di valutazione usata è la seguente:

Uso dei costrutti di Jolie	Corretto utilizzo dei costrutti per la gestione del parallelismo, uso di `execution(concurrent)` a dispetto dell'uso di `execution(sequential)` in tutti i servizi.
Distribuzione del carico di lavoro nel gruppo	Omogeneità nella ripartizione dei compiti nel gruppo, ogni membro partecipa egualmente allo sviluppo, assenza di dissimmetria di informazione.
Grado di partecipazione alla comunità	Presenza di domande e risposte sulla newsgroup del corso

3.3 La Demo

La demo, o dimostrazione di funzionamento, esegue e monitora il comportamento di un sistema di gestione decentralizzata di transazioni. Essa simula almeno quattro utenti che entrano a far parte della rete peer-to-peer come nodi, un server timestamp che gestisce la sincronizzazione oraria ed un tool per monitorare lo stato della rete (il Network Visualiser). La sequenza di esecuzione è la seguente:

1. avvio del Server Timestamp;
2. avvio del Network Visualizer;
3. avvio del primo nodo, generatore del primo blocco (con *reward* di sei Jollar);
4. avvio del secondo, terzo e quarto nodo (con conseguente *download* della blockchain);
5. invio di un Jollar dal primo al secondo nodo (con conseguente scrittura su un nuovo blocco e reward relativo);
6. invio di due Jollar dal primo al terzo nodo (con conseguente scrittura su un nuovo blocco e reward relativo);
7. invio di tre Jollar dal primo al quarto nodo (con conseguente scrittura su un nuovo blocco e reward relativo);
8. richiesta al Network Visualiser di stampare la situazione relativa al totale dei Jollar presenti nella rete e dei relativi possessori, con l'elenco delle transazioni avvenute.