

# Jollar – Laboratorio Sistemi Operativi A.A. 2017-2018

Per un Pugno di Jollar

[giacomo.minello@studio.unibo.it](mailto:giacomo.minello@studio.unibo.it)

September 12, 2018

## Abstract

Questo documento ha lo scopo di essere un report dettagliato sulle modalità di esecuzione e le scelte implementative della blockchain Jollar. Verranno indicati i contatti dei membri del gruppo, si procederà quindi ad una breve descrizione delle caratteristiche del progetto. Seguono le istruzioni ed i comandi di esecuzione per una demo che è stata preparata. Infine si discuteranno le strategie implementative con particolare riguardo all'architettura dei servizi.

## Indice

<b>1</b>	<b>Contatti</b>	<b>2</b>
<b>2</b>	<b>Descrizione del progetto</b>	<b>3</b>
<b>3</b>	<b>Istruzioni per la demo</b>	<b>4</b>
<b>4</b>	<b>Discussione delle strategie di implementazione</b>	<b>5</b>
4.1	Organizzazione . . . . .	5
4.2	Struttura del progetto . . . . .	5
4.2.1	Struttura di un nodo . . . . .	5
4.2.2	Struttura di main.ol . . . . .	6
4.2.3	Struttura del Network Visualizer . . . . .	6
4.2.4	Struttura della demo . . . . .	6

# 1 Contatti

## Contatti del gruppo

- Minello Giacomo<sup>a</sup>, Matr. 0000802402,  
[giacomo.minello@studio.unibo.it](mailto:giacomo.minello@studio.unibo.it)  
[minellogiacomo@gmail.com](mailto:minellogiacomo@gmail.com)
- Aspromonte Marco, Matr. 0000806519  
[marco.aspromonte@studio.unibo.it](mailto:marco.aspromonte@studio.unibo.it)  
[marco.aspromonte@gmail.com](mailto:marco.aspromonte@gmail.com)
- Menetto Davide, Matr. 0000768828  
[davide.menetto@studio.unibo.it](mailto:davide.menetto@studio.unibo.it)  
[davide.menetto96@gmail.com](mailto:davide.menetto96@gmail.com)
- Morselli Enrico, Matr. 0000806725  
[enrico.morselli@studio.unibo.it](mailto:enrico.morselli@studio.unibo.it)
- Memoli Alessandro, Matr. 0000

---

<sup>a</sup>Referente del gruppo

## **2    Descrizione del progetto**

Il progetto assegnato prevede la creazione di una blockchain, ovvero il codice dei nodi che partecipano ad una rete P2P per lo scambio e la verifica di transazioni basate sulla cryptocurrency Jollar.

### 3 Istruzioni per la demo

La sequenza di esecuzione è la seguente:

1. avvio del Nodo 1;  
jolie node1.ol
2. avvio del Nodo 2;  
jolie node2.ol
3. avvio del Nodo 3;  
jolie node3.ol
4. avvio del Nodo 4;  
jolie node4.ol
5. avvio del Network Visualizer  
jolie networkVisualizer.ol
6. avvio di DemoTX.ol  
jolie demoTX.ol

## 4 Discussione delle strategie di implementazione

In questa sezione sono analizzate la struttura del progetto, le scelte e i vincoli implementativi e la descrizione specifica delle singole feature.

### 4.1 Organizzazione

### 4.2 Struttura del progetto

#### 4.2.1 Struttura di un nodo

Riguardo alla struttura del codice è bene iniziare analizzando la struttura di un nodo.

```
constants {  
  ROOT="socket://localhost:900",  
  CREATEGENESISBLOCK = true,  
  ID="1",  
  LOCATION="socket://localhost:9001"  
}  
include "main.ol"
```

Si osserva quindi che, contrariamente a quanto ci si aspetti, un file nodoN.ol contiene una piccola porzione di codice. Per massimizzare la code reusability e per agevolare la mantenibilità del codice abbiamo scelto di identificare un nodo con quattro costanti (di cui solo tre caratteristiche di ogni nodo). Queste costanti sono in ordine:

- ROOT, costante comune a tutti i nodi, è la radice di una location ed è stata inserita per evitare di avere ogni indirizzo hardcoded nel codice.
- CREATEGENESISBLOCK, è una costante booleana con valore true solo per il primo nodo della rete, il suo scopo si può dedurre dal nome.
- ID, questo valore corrisponde al numero n del nodoN, viene usato per identificare univocamente un nodo.
- LOCATION, indica l'indirizzo dell'inputPort del nodo, è stata inserita solo perché il linguaggio Jolie non permette di dichiarare una location tramite l'operazione sulle stringhe ROOT+ID.

Ogni nodo importa il file "main.ol" che contiene il codice comune a tutti i nodi.

#### **4.2.2 Struttura di main.ol**

Il file main.ol di cui riporteremo qui solo alcuni estratti per ragioni di chiarezza e lunghezza si occupa di importare alcune librerie standard del linguaggio Jolie per poter esporre al programma alcuni servizi utili. Per ragioni di compattezza abbiamo scelto di impostare solo due porte per ogni nodo, una di output e una di input. Per fare ciò abbiamo deciso di considerare la location dell'inputPort come univoca per ogni nodo e dichiarare la location dell'output port tramite dynamic binding. ectna di ionDescrizione delle feature

#### **4.2.3 Struttura del Network Visualizer**

#### **4.2.4 Struttura della demo**