



PineSU -
Workflow

Paolo Speziali

Operazioni

Creazione

Registrazione

Esportazione

Controllo file
esportati

Controllo
d'integrità su
una SU

Workflow

Resoconto

PineSU - Workflow

Sviluppo di un'applicazione distribuita su Blockchain
Ethereum

Paolo Speziali

Università degli Studi di Perugia - Dipartimento di Ingegneria



A.A. 2020/2021



Operazioni eseguibili

PineSU -
Workflow

Paolo Speziali

Operazioni

Creazione

Registrazione

Esportazione

Controllo file
esportati

Controllo
d'integrità su
una SU

Workflow

Resoconto

Nell'applicativo è possibile eseguire le seguenti operazioni:

- 1 Creazione di una Storage Unit (SU)
- 2 Registrazione di una o più SU nella blockchain Ethereum
- 3 Esportazione di sottoinsiemi di file da una SU
- 4 Controllo d'integrità di singoli file esportati da altre SU
- 5 Controllo d'integrità su una SU

Oltre a poter eseguire tutti i comandi di Git. Vedremo ora questi comandi uno per uno utilizzando una directory campione da trasformare in Storage Unit.



Directory campione

PineSU -
Workflow

Paolo Spezioli

Operazioni

Creazione

Registrazione

Esportazione

Controllo file
esportati

Controllo
d'integrità su
una SU

Workflow

Resoconto

Utilizzeremo come directory campione per mostrare l'evoluzione della Storage Unit una cartella chiamata *sample* con la seguente struttura (dove *first* e *second* sono sottodirectory di *sample* e *third* è sottodirectory di *second*):

```
sample
sample/first
sample/first/astar.js
sample/first/graph.js
sample/graphCreator.js
sample/second
sample/second/priorityQueue.js
sample/second/third
sample/second/third/main.js
sample/second/third/vertex.js
```



Creazione di una SU

PineSU -
Workflow

Paolo Speciali

Operazioni

Creazione

Registrazione

Esportazione

Controllo file
esportati

Controllo
d'integrità su
una SU

Workflow

Resoconto

La creazione di una Storage Unit è in realtà un'operazione che comprende sia la trasformazione in una Git Repository della directory, sia il calcolo degli hash che serviranno poi per registrare la nostra SU nella blockchain.

Le informazioni della nostra SU sono conservate nel file **.pinesu.json** nella root della directory, la presenza di questo file indica al programma che la directory è già una SU.

La creazione potrebbe eventualmente venire suddivisa in due azioni diverse, ma a quel punto la prima fase si limiterebbe ad eseguire un comando "git init" sulla directory, comando che potrebbe non essere necessario.



Creazione di una SU

PineSU -
Workflow

Paolo Speziali

Operazioni

Creazione

Registrazione

Esportazione

Controllo file
esportati

Controllo
d'integrità su
una SU

Workflow

Resoconto

Tuttavia, per come il programma è ora strutturato, la creazione della Storage Unit la rende pronta per essere registrata (eventualmente insieme ad altre) nella blockchain.

Andiamo quindi a vedere quali sono le operazioni che avvengono in questa fase e come la nostra directory *sample* cambia.



Creazione di una SU

PineSU -
Workflow

Paolo Speziali

Operazioni

Creazione

Registrazione

Esportazione

Controllo file
esportati

Controllo
d'integrità su
una SU

Workflow

Resoconto

- Controlla se nella directory è già presente **.pinesu.json**, in caso negativo continua;
- Permette all'utente di selezionare dei file da ignorare da tutto il processo, ciò produce un file **.gitignore**;
- Aggiunge tutti i file della directory nella Git Staging Area e crea un commit fantoccio;
- Recupera la lista dei file dal commit;
- Calcola gli hash di tutti i file (non directory);
- Calcola gli hash delle directory creando dei Merkle Tree (MT) con gli hash del loro contenuto e usa l'hash della radice come hash della directory, ciò avviene tramite un approccio bottom-up: calcola prima gli hash delle directory senza file al loro interno;



Creazione di una SU

PineSU -
Workflow

Paolo Speziali

Operazioni

Creazione

Registrazione

Esportazione

Controllo file
esportati

Controllo
d'integrità su
una SU

Workflow

Resoconto

- Calcola l'hash della SU creando un Merkle Tree con tutti gli hash dei file e delle directory in essa contenute e ne prende la radice;
- Viene generato il file **.pinesu.json** con al suo interno nome, descrizione, visibilità, data, hash dell'utente, hash della SU, lista dei file e delle sottodirectory con hash associati;
- Aggiunge tutti i file della directory nella Git Staging Area e crea un commit fantoccio;
- Aggiungo il percorso di questa SU con il suo hash alla lista della SU generate dall'utente (memorizzata nella cartella d'installazione del programma);
- Viene annullato il commit fantoccio e creato uno effettivo con informazioni consistenti e aggiunto **.pinesu.json**;



Creazione di una SU - Risultato

PineSU -
Workflow

Paolo Speziali

Operazioni

Creazione

Registrazione

Esportazione

Controllo file
esportati

Controllo
d'integrità su
una SU

Workflow

Resoconto

L'operazione di creazione ha prodotto nella cartella *sample*, ovviamente, una sottocartella *.git* e il file **.pinesu.json**.

L'ordine di calcolo degli hash è stato questo:

- 1 sample/first/astar.js
- 2 sample/first/graph.js
- 3 sample/graphCreator.js
- 4 sample/second/priorityQueue.js
- 5 sample/second/third/main.js
- 6 sample/second/third/vertex.js



Creazione di una SU - Risultato

PineSU -
Workflow

Paolo Speziali

Operazioni

Creazione

Registrazione

Esportazione

Controllo file
esportati

Controllo
d'integrità su
una SU

Workflow

Resoconto

- 7 sample/first (tramite MT dei due file da lui contenuti)
- 8 sample/second/third (tramite MT dei due file da lui contenuti)
- 9 sample/second (tramite MT del file e della sottodirectory *third* da lui contenuti)
- 10 sample (tramite MT degli hash dei tutti i file e le directory appena visti)

Si può quindi facilmente osservare come questa tecnica di calcolo di hash bottom-up viene messa in atto calcolando prima i file effettivi, poi le directory senza sottodirectory e infine tutto il resto delle directory.



Chiarificazione

PineSU -
Workflow

Paolo Speziali

Operazioni

Creazione

Registrazione

Esportazione

Controllo file
esportati

Controllo
d'integrità su
una SU

Workflow

Resoconto

L'hash che verrà infine inserito nella blockchain non sarà tuttavia l'hash della directory *sample*, bensì l'hash del suo file **.pinesu.json**, ogni volta quindi che farò riferimento all'hash della SU mi starò riferendo all'hash del file JSON associato.



Registrazione di una o più SU nella blockchain

PineSU -
Workflow

Paolo Spezioli

Operazioni

Creazione

Registrazione

Esportazione

Controllo file
esportati

Controllo
d'integrità su
una SU

Workflow

Resoconto

In questa fase all'utente viene data la libertà di scegliere se registrare con un unico hash una o più SU insieme, poiché il workflow è praticamente identico andrò a separare i punti esclusivi ad una registrazione “multipla” scrivendoli in color oliva:

- Mostra la selezione multipla della SU registrate dall'utente per permettere di registrarne una sola o più in una volta;
- Per ogni Storage Unit selezionata creo nella root della SU il file **.registration.json** che contiene l'hash della directory, l'hash che verrà salvato nella blockchain (stesso valore se registriamo una sola SU, **valore della root del MT di tutti gli hash delle SU selezionate se la registrazione è multipla**)



Registrazione di una o più SU nella blockchain

PineSU -
Workflow

Paolo Speciali

Operazioni

Creazione

Registrazione

Esportazione

Controllo file
esportati

Controllo
d'integrità su
una SU

Workflow

Resoconto

- Nel caso di una registrazione multipla, va a calcolare ed inserire in **.registration.json** le *proof*, ovvero le informazioni che permettono di risalire all'hash registrato su blockchain dato l'hash della singola SU;
- Apre una scheda del browser per permettere il pagamento in ETH, nella blockchain sarà registrato l'hash della SU (o **risultante da più SU**) unito all'hash personale dell'utente della registrazione (non si esclude infatti la possibilità di implementare un'autenticazione centralizzata).



Esportazione di sottoinsiemi di file da una SU

PineSU -
Workflow

Paolo Speziali

Operazioni

Creazione

Registrazione

Esportazione

Controllo file
esportati

Controllo
d'integrità su
una SU

Workflow

Resoconto

In questa fase all'utente viene data la possibilità di scegliere di esportare alcuni file dalla SU, essi potranno poi in seguito essere integrati in altre SU e ne potrà essere controllata l'integrità:

- Mostra la selezione multipla dei file della SU per permettere all'utente di scegliere quali esportare;
- Viene creato un file **.pifiles.json** in cui salva, per ogni file esportato, il suo percorso originale, il suo hash, l'hash della root e le *proof* per calcolare la root dato l'hash del file (ciò servirà nell'operazione di verifica d'integrità);
- I file, seguendo la struttura in cui comparivano nella SU originale, e **.pifiles.json** vengono compressi in un file ZIP (chiamato **pinesuExport.zip**) e salvati nella cartella precedente a quella in cui si sta operando;



Esportazione di sottoinsiemi di file da una SU - Risultato

PineSU -
Workflow

Paolo Speziali

Operazioni

Creazione

Registrazione

Esportazione

Controllo file
esportati

Controllo
d'integrità su
una SU

Workflow

Resoconto

L'operazione di esportazione dei file *sample/graphCreator.js*, *sample/first/graph.js* e *sample/second/third/main.js* ha prodotto un file **pinesuExport.zip** contenente i file esportati e la struttura delle cartelle mantenuta inalterata (è presente un bug che taglia i nomi delle cartelle nel file ZIP, nulla di grave ma sto comunque investigando sulla causa di ciò) e il file **.pifiles.json**.



Controllo d'integrità di singoli file esportati da altre SU

PineSU -
Workflow

Paolo Speziali

Operazioni

Creazione

Registrazione

Esportazione

Controllo file
esportati

Controllo
d'integrità su
una SU

Workflow

Resoconto

L'unica operazione svolta dal controllo d'integrità dei file esportati è quella di cercare i **.pifiles.json** nella cartella d'interesse e controllare che l'hash di ogni file combaci con quello registrato nel documento JSON e che si riesca a risalire all'hash della SU originale tramite le *proof* registrate sempre nel medesimo JSON.



Controllo d'integrità su una SU

PineSU -
Workflow

Paolo Speziali

Operazioni

Creazione

Registrazione

Esportazione

Controllo file
esportati

Controllo
d'integrità su
una SU

Workflow

Resoconto

In questa operazione l'applicativo svolge un controllo d'integrità ricalcolando gli hash, confrontandoli con i dati registrati in precedenza e controllando la presenza dell'hash su blockchain.

- Avviene un ricalcolo degli hash di file e directory come descritto nella fase di creazione;
- Si legge il file **.pinesu.json** e si confrontano i dati appena calcolati con quelli vecchi;
- Si richiama l'operazione di controllo dei file integrati da altre SU;
- In caso i dati in **.pinesu.json** combacino con i dati appena calcolati si procede a controllare la presenza dell'hash della SU su blockchain, ciò avviene leggendo **.registration.json**, verificando le *proof* del MT in caso di registrazione multipla e cercando nella blockchain i dati richiesti.



Workflow di una Storage Unit

PineSU -
Workflow

Paolo Speziali

Operazioni

Creazione

Registrazione

Esportazione

Controllo file
esportati

Controllo
d'integrità su
una SU

Workflow

Resoconto

Nell'immagine della slide successiva è possibile osservare il riassunto, in forma grafica, dei concetti delle slide precedenti. I sensi delle frecce permettono di capire quali azioni cambiano lo stato di una Storage Unit e quali azioni sono irreversibili (almeno all'interno dell'applicativo).



Workflow di una Storage Unit

PineSU -
Workflow

Paolo Speziali

Operazioni

Creazione

Registrazione

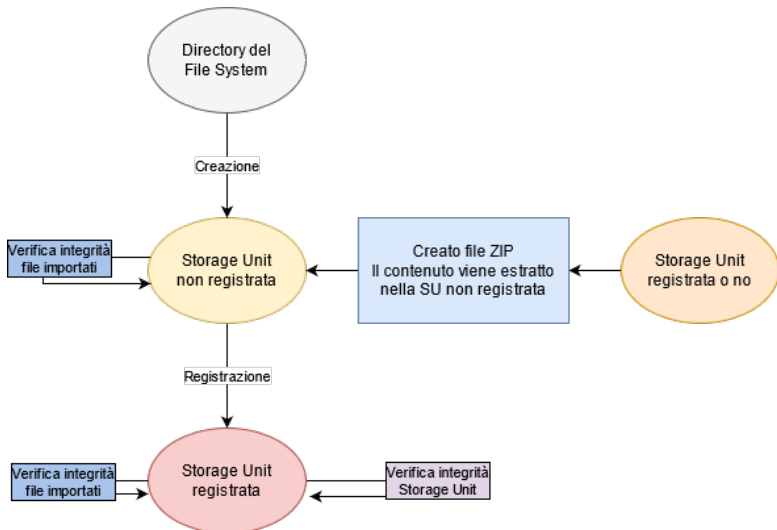
Esportazione

Controllo file
esportati

Controllo
d'integrità su
una SU

Workflow

Resoconto





Resoconto

PineSU -
Workflow

Paolo Speziali

Operazioni

Creazione

Registrazione

Esportazione

Controllo file
esportati

Controllo
d'integrità su
una SU

Workflow

Resoconto

- Professore Tutor: Luca Grilli
- Studente Tirocinante: Paolo Speziali
- Anno Accademico: 2020/2021