



Condividere  
informazioni  
in modo  
sicuro  
combinando  
Git e  
Blockchain

Laureando:  
Paolo Speciali  
Relatore:  
Luca Grilli

Il Problema

L'Obiettivo

Il Software  
PineSU

Tecnologie  
utilizzate

Sviluppi futuri

# Condividere informazioni in modo sicuro combinando Git e Blockchain

Laureando: Paolo Speciali  
Relatore: Luca Grilli

Università degli Studi di Perugia - Dipartimento di Ingegneria  
Corso di laurea triennale in Ingegneria Informatica ed Elettronica



A.D. 1308  
**unipg**  
DIPARTIMENTO  
DI INGEGNERIA

A.A. 2020/2021



# Indice

Condividere  
informazioni  
in modo  
sicuro  
combinando  
Git e  
Blockchain

Laureando:  
Paolo Speziali  
Relatore:  
Luca Grilli

Il Problema

L'Obiettivo

Il Software  
PineSU

Tecnologie  
utilizzate

Sviluppi futuri

## 1 Il Problema

## 2 L'Obiettivo

## 3 Il Software PineSU

## 4 Tecnologie utilizzate

## 5 Sviluppi futuri



# La digitalizzazione

Condividere  
informazioni  
in modo  
sicuro  
combinando  
Git e  
Blockchain

Laureando:  
Paolo Speciali  
Relatore:  
Luca Grilli

Il Problema

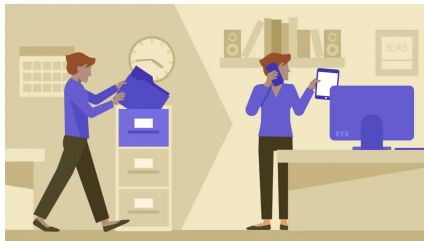
L'Obiettivo

Il Software  
PineSU

Tecnologie  
utilizzate

Sviluppi futuri

È in atto, negli ultimi anni, un piano di **digitalizzazione** della **Pubblica Amministrazione**. L'esigenza di questa trasformazione si è fatta sentire anche da parte dell'**Unione Europea**, che con il **Recovery Fund** ci sta fornendo i fondi per attuarla, ben **11,75 milioni di euro**.





# Il problema della burocrazia

Condividere  
informazioni  
in modo  
sicuro  
combinando  
Git e  
Blockchain

Laureando:  
Paolo Speciali  
Relatore:  
Luca Grilli

Il Problema

L'Obiettivo

Il Software  
PineSU

Tecnologie  
utilizzate

Sviluppi futuri

**Tab. 2 – Stima costo annuo burocrazia sulle imprese per regione (\*)**

Rank	Regioni e ripartizioni	Valore aggiunto (mln €)	Inc. % VA su totale Italia	Stima costo annuo sostenuto dalle imprese per la gestione dei rapporti con la PA (mln €)
1	Lombardia	343.840	22,1	12.625
2	Lazio	176.024	11,3	6.463
3	Veneto	143.221	9,2	5.259
4	Emilia-Romagna	141.373	9,1	5.191
5	Piemonte	120.689	7,7	4.431
6	Toscana	102.735	6,6	3.772
7	Campania	96.682	6,2	3.550
8	Sicilia	79.274	5,1	2.911
9	Puglia	67.279	4,3	2.470
10	Liguria	44.027	2,8	1.617
11	Trentino Alto Adige	39.651	2,5	1.456
12	Marche	37.315	2,4	1.370
13	Friuli-Venezia Giulia	33.540	2,2	1.232
14	Sardegna	30.561	2,0	1.122
15	Calabria	29.886	1,9	1.097
16	Abruzzo	29.392	1,9	1.079
17	Umbria	19.959	1,3	733
18	Basilicata	11.139	0,7	409
19	Molise	5.654	0,4	208
20	Valle d'Aosta	4.283	0,3	157
<b>ITALIA</b>		<b>1.557.833</b>	<b>100,0</b>	<b>57.200</b>
	Nord Ovest	512.839	32,9	18.830
	Nord Est	357.784	23,0	13.137
	Centro	336.032	21,6	12.338
	Mezzogiorno	349.866	22,5	12.846
	Extra-Regio	1.312	0,1	48

Elaborazione Ufficio Studi CGIA su dati The European House Ambrosetti e Istat

(\*) Stima costruita utilizzando dati 2017 applicando la ripartizione del valore aggiunto a livello territoriale.

Serve una **sburo-  
cratizzazione**  
grazie a software  
per **salvare**,  
**autenticare** e  
**condividere**  
documenti in  
maniera **sicura**.



# Stato dell'arte

Condividere  
informazioni  
in modo  
sicuro  
combinando  
Git e  
Blockchain

Laureando:  
Paolo Speciali  
Relatore:  
Luca Grilli















Il Problema

L'Obiettivo

Il Software  
PineSU

Tecnologie  
utilizzate

Sviluppi futuri

Proprietà	Version Control System	Blockchain
Condivis.		
Tracciabil.		
Autenticità		
Integrità		
Efficienza		
Flessibilità		
Costi		



# Git

Condividere  
informazioni  
in modo  
sicuro  
combinando  
Git e  
Blockchain

Laureando:  
Paolo Speciali  
Relatore:  
Luca Grilli

## Il Problema

### L'Obiettivo

Il Software  
PineSU

Tecnologie  
utilizzate

Sviluppi futuri

**Git** è il sistema di controllo di versione (**VCS**) distribuito più diffuso al mondo.

Esso agevola la gestione **distribuita** di insiemi di file e directory. Un VCS considera tali insiemi unità chiamate **repository**.

Git ci permette di:

- **Tracciare** le modifiche in una repository.
- **Ripristinare** le repository ad uno stato precedente.
- **Condividere** le repository con il loro storico dei cambiamenti.

e molto altro...





# Git - Un po' di numeri

Condividere  
informazioni  
in modo  
sicuro  
combinando  
Git e  
Blockchain

Laureando:  
Paolo Speciali  
Relatore:  
Luca Grilli

Il Problema


L'Obiettivo

Il Software  
PineSU

Tecnologie  
utilizzate

Sviluppi futuri

Showing 63,289,757 available users ⓘ

-  **Linus Torvalds** torvalds  
Portland, OR
-  **Ruan YiFeng** ruanyf  
Shanghai, China yifeng.ruan@gmail.com
-  **Jake Wharton** JakeWharton  
Pittsburgh, PA, USA j@ke.fyi
-  **Addy Osmani** addyosmani

🔍 Search more than 238M repositories

Search GitHub

ProTip! For an advanced search, use some of our prefixes.

## Companies & Projects Using Git

Google

FACEBOOK

Microsoft



LinkedIn

NETFLIX





# Blockchain

Condividere  
informazioni  
in modo  
sicuro  
combinando  
Git e  
Blockchain

Laureando:  
Paolo Speziali  
Relatore:  
Luca Grilli

Il Problema

L'Obiettivo

Il Software  
PineSU

Tecnologie  
utilizzate

Sviluppi futuri

La **blockchain** è un registro in continua crescita di record chiamati **blocchi**, collegati l'uno all'altro come in una **catena** grazie a **metodi crittografici**. Essa è:

- **Immutabile.**
- **Distribuita.**
- **Estremamente sicura.**







# Blockchain

Condividere  
informazioni  
in modo  
sicuro  
combinando  
Git e  
Blockchain

Laureando:  
Paolo Speziali  
Relatore:  
Luca Grilli

Il Problema

L'Obiettivo

Il Software  
PineSU

Tecnologie  
utilizzate

Sviluppi futuri

La **blockchain** è un registro in continua crescita di record chiamati **blocchi**, collegati l'uno all'altro come in una **catena** grazie a **metodi crittografici**. Essa è:

- **Immutabile.**
- **Distribuita.**
- **Estremamente sicura.**



È alla base delle reti di criptovalute, come **Ethereum**, su cui si possono anche costruire applicazioni decentralizzate con gli **Smart Contract**.



# L'Obiettivo

Condividere  
informazioni  
in modo  
sicuro  
combinando  
Git e  
Blockchain

Laureando:  
Paolo Speciali  
Relatore:  
Luca Grilli

Il Problema

L'Obiettivo

Il Software  
PineSU

Tecnologie  
utilizzate

Sviluppi futuri

Realizzare uno **software** che combini **Git** con una **blockchain** e sia in grado di:



**Salvare** hash di  
repository su  
blockchain



**Esportare**  
sottoinsiemi di  
repository verificabili



**Verificare** l'integrità  
di singoli file e  
repository



# Autenticazione con blockchain

Condividere  
informazioni  
in modo  
sicuro  
combinando  
Git e  
Blockchain

Laureando:  
Paolo Speciali  
Relatore:  
Luca Grilli

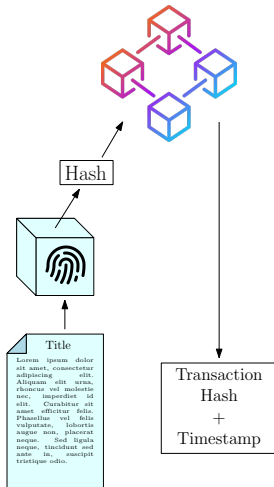
Il Problema

L'Obiettivo

Il Software  
PineSU

Tecnologie  
utilizzate

Sviluppi futuri



- 1 Associazione **hash** al documento tramite una **funzione crittografica di hashing**.
- 2 Registrazione dell'hash su un **blocco** della blockchain.
- 3 Restituzione e salvataggio dell'**hash della transazione** e del **timestamp**.



# Funzioni crittografiche di hashing

Condividere  
informazioni  
in modo  
sicuro  
combinando  
Git e  
Blockchain

Laureando:  
Paolo Speziali  
Relatore:  
Luca Grilli

Il Problema

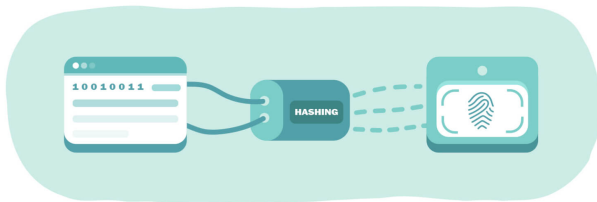
L'Obiettivo

Il Software  
PineSU

Tecnologie  
utilizzate

Sviluppi futuri

Funzione che **associa**, a una qualsiasi sequenza  $m$  di lunghezza arbitraria in input, una sequenza in output  $h(m)$  di lunghezza costante, seguendo alcune proprietà che la rendono *crittograficamente sicura*. Ciò impedisce di risalire all'input originale e facilita i **controlli di integrità sui file**.





# Il problema della blockchain

Condividere  
informazioni  
in modo  
sicuro  
combinando  
Git e  
Blockchain

Laureando:  
Paolo Speziali  
Relatore:  
Luca Grilli

Il Problema

L'Obiettivo

Il Software  
PineSU

Tecnologie  
utilizzate

Sviluppi futuri

Non possiamo però utilizzare un blocco per registrare **un solo file** e nemmeno **una sola repository**, sarebbe troppo **costoso**. La soluzione è l'utilizzo di **accumulatori crittografici**: strumenti che **comprimono molte informazioni** in una **costante** di dimensione fissa.



# Il Software PineSU

Condividere  
informazioni  
in modo  
sicuro  
combinando  
Git e  
Blockchain

Laureando:  
Paolo Speciali  
Relatore:  
Luca Grilli

Il Problema

L'Obiettivo

Il Software  
PineSU

Tecnologie  
utilizzate

Sviluppi futuri

**PineSU** è un software **Javascript** che sfrutta il run-time **Node.js**.

L'applicazione crea delle **strutture** sulle repository Git chiamate **Storage Unit (SU)** tramite metadati.

Queste SU sono le unità su cui effettueremo le singole operazioni, eccetto la registrazione su blockchain che si svolgerà collettivamente con l'ausilio di accumulatori crittografici.





# Storage Unit

Condividere  
informazioni  
in modo  
sicuro  
combinando  
Git e  
Blockchain

Laureando:  
Paolo Speciali  
Relatore:  
Luca Grilli

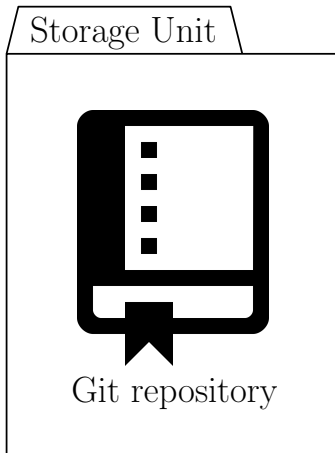
Il Problema

L'Obiettivo

Il Software  
PineSU

Tecnologie  
utilizzate

Sviluppi futuri



Metadati:

- Nome;
- Link repo remota;
- Descrizione;
- Nome;
- Visibilità;
- Data;
- Lista hash;
- Chiusura;



# Workflow

Condividere  
informazioni  
in modo  
sicuro  
combinando  
Git e  
Blockchain

Laureando:  
Paolo Speziali  
Relatore:  
Luca Grilli

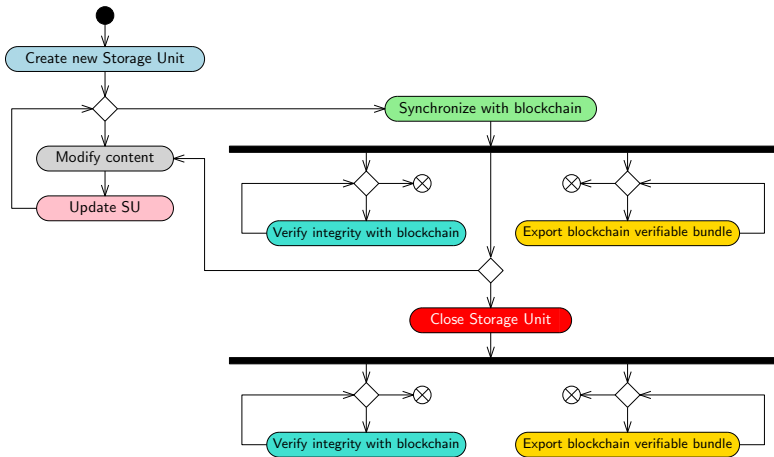
Il Problema

L'Obiettivo

Il Software  
PineSU

Tecnologie  
utilizzate

Sviluppi futuri







# Ciclo vitale di una Storage Unit

Condividere  
informazioni  
in modo  
sicuro  
combinando  
Git e  
Blockchain

Laureando:  
Paolo Speziali  
Relatore:  
Luca Grilli

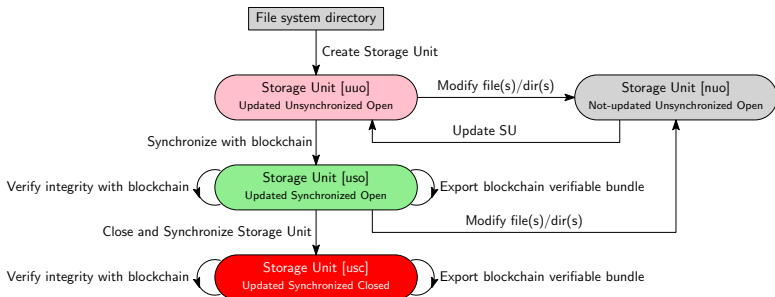
Il Problema

L'Obiettivo

Il Software  
PineSU

Tecnologie  
utilizzate

Sviluppi futuri





# Architettura

Condividere  
informazioni  
in modo  
sicuro  
combinando  
Git e  
Blockchain

Laureando:  
Paolo Speziali  
Relatore:  
Luca Grilli

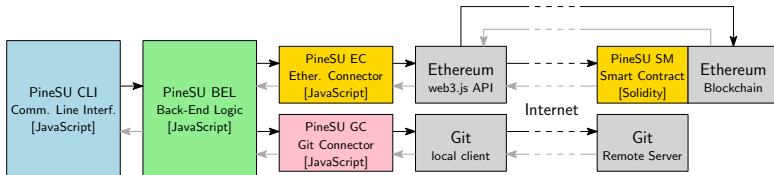
Il Problema

L'Obiettivo

Il Software  
PineSU

Tecnologie  
utilizzate

Sviluppi futuri





# Architettura (Cont.)

Condividere  
informazioni  
in modo  
sicuro  
combinando  
Git e  
Blockchain

Laureando:  
Paolo Speziali  
Relatore:  
Luca Grilli

Il Problema

L'Obiettivo

Il Software  
PineSU

Tecnologie  
utilizzate

Sviluppi futuri

- **PineSU CLI** (*Command Line Interface*): **Crea l'interfaccia utente** con cui è possibile interagire e **richiama le funzioni** degli altri moduli all'occorrenza.
- **PineSU BEL** (*Back End Logic*): Il **nucleo** di PineSU. **Gestisce le SU** e controlla la comunicazione con la **blockchain** e il client **Git** locale.
- **PineSU EC** (*Ethereum Connector*): Si interfaccia con le **API della blockchain**.
- **PineSU GC** (*Git Connector*): Si interfaccia con il **client Git**.
- **PineSU SM** (*Smart Contract*): Permette **registrazioni permanenti** di singole SU nella blockchain.



# PineSU CLI

Condividere  
informazioni  
in modo  
sicuro  
combinando  
Git e  
Blockchain

Laureando:  
Paolo Speciali  
Relatore:  
Luca Grilli

Il Problema

L'Obiettivo

Il Software  
PineSU

Tecnologie  
utilizzate

Sviluppi futuri

```
MINGW64:/d/Progetti/Tesi

PineSU

? Welcome to PineSU, choose the operation to perform (Use arrow keys)
> Create new SU / Recalculate open SU
  Stage Storage Unit for Synchronization
  Close current SU
  Register Staged SUs in the blockchain network
  Check SU integrity
  Export files from current SU
  Check files integrity
(Move up and down to reveal more choices)
```



Condividere  
informazioni  
in modo  
sicuro  
combinando  
Git e  
Blockchain

Laureando:  
Paolo Speziali  
Relatore:  
Luca Grilli

Il Problema

L'Obiettivo

Il Software  
PineSU

Tecnologie  
utilizzate

Sviluppi futuri

Il nucleo centrale che si occupa di:

- 1 **Gestione dei file descrittori.**
- 2 **Gestione degli accumulatori crittografici.**
- 3 **Comunicazione con Git e blockchain.**



# Merkle Calendar

Condividere  
informazioni  
in modo  
sicuro  
combinando  
Git e  
Blockchain

Laureando:  
Paolo Speciali  
Relatore:  
Luca Grilli

Il Problema

L'Obiettivo

Il Software  
PineSU

Tecnologie  
utilizzate

Sviluppi futuri

Il *Merkle Calendar (MC)* è l'**accumulatore critografico** più importante di PineSU. Si tratta di un albero in cui le **foglie** sono i Blockchain Synchronization Point (**BSP**), istanze di Storage Group, a loro volta raggruppate in nodi rappresentanti **mesi e anni**, ciò rende i reperimenti di registrazioni passate più agevoli e veloci.

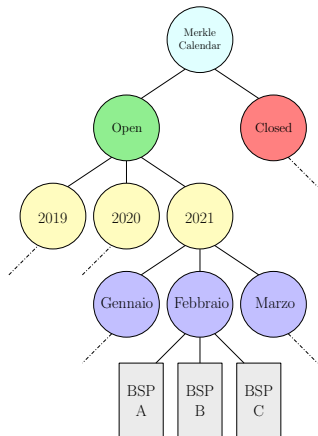


Figura: Un Merkle Calendar



# Merkle Calendar UML

Condividere  
informazioni  
in modo  
sicuro  
combinando  
Git e  
Blockchain

Laureando:  
Paolo Speciali  
Relatore:  
Luca Grilli

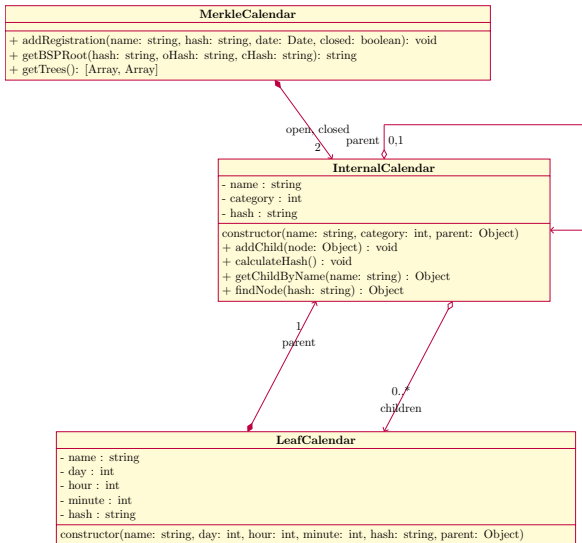
Il Problema

L'Obiettivo

Il Software  
PineSU

Tecnologie  
utilizzate

Sviluppi futuri





# Codice - Reperimento di una BSP Root

Condividere  
informazioni  
in modo  
sicuro  
combinando  
Git e  
Blockchain

Laureando:  
Paolo Speciali  
Relatore:  
Luca Grilli

Il Problema

L'Obiettivo

Il Software  
PineSU

Tecnologie  
utilizzate

Sviluppi futuri

```
for(let i = 0; i <= leafIndex; i++){
    leavesHash.push(monthNode.getChildByNum(i).getHash())
}
let newMonth = this.calculateHash(leavesHash);
let monthsHash = new Array();
for(let i = 0; i < monthIndex; i++){
    monthsHash.push(yearNode.getChildByNum(i).getHash())
}
monthsHash.push(newMonth);
let newYear = this.calculateHash(monthsHash);
let yearsHash = new Array();
for(let i = 0; i < yearIndex; i++){
    yearsHash.push(yearNode.getChildByNum(i).getHash())
}
yearsHash.push(newYear);
let newRoot = this.calculateHash(yearsHash)
```





Condividere  
informazioni  
in modo  
sicuro  
combinando  
Git e  
Blockchain

Laureando:  
Paolo Speziali  
Relatore:  
Luca Grilli

Il Problema

L'Obiettivo

Il Software  
PineSU

Tecnologie  
utilizzate

Sviluppi futuri

## GitConnector

- git : SimpleGit

```
constructor(dir: string)
+ init() : void
+ add(arg: string) : void
+ commit(msg: string, enmsg: boolean) : void
+ getRepoFiles() : Array
+ push() : void
+ pull() : void
+ reset() : void
+ hasRemote() : Array
+ custom(commands: Array) : string
```



Condividere  
informazioni  
in modo  
sicuro  
combinando  
Git e  
Blockchain

Laureando:  
Paolo Speciali  
Relatore:  
Luca Grilli

Il Problema

L'Obiettivo

Il Software  
PineSU

Tecnologie  
utilizzate

Sviluppi futuri

## EthConnector

- web3 : Web3
- w1 : string
- w2 : string
- k : string

```
constructor(host: string, w1 : string, w2 : string, k : string)
+ deploy(hashRoot: string) : string
+ verifyHash(transHash: string, hash: string) : boolean
```



# Codice - Salvataggio di un hash su blockchain

Condividere  
informazioni  
in modo  
sicuro  
combinando  
Git e  
Blockchain

Laureando:  
Paolo Speciali  
Relatore:  
Luca Grilli

Il Problema

L'Obiettivo

Il Software  
PineSU

Tecnologie  
utilizzate

Sviluppi futuri

```
async deploy(hashRoot){
  const ct = await this.web3.eth.accounts
    .signTransaction({
      from: this.w1,
      to: this.w2,
      data: hashRoot,
      gas: 3000000,
    },
    this.k
  );
  const receipt = await this.web3.eth
    .sendSignedTransaction(ct.rawTransaction);
  return receipt.transactionHash;
}
```



# PineSU SM

Condividere  
informazioni  
in modo  
sicuro  
combinando  
Git e  
Blockchain

Laureando:  
Paolo Speziali  
Relatore:  
Luca Grilli

Il Problema

L'Obiettivo

Il Software  
PineSU

Tecnologie  
utilizzate

Sviluppi futuri

```
contract SURegistry {  
  
    string StorageUnit;  
    mapping(uint => string) public registry;  
    uint public SUCount;  
  
    function addSU(string memory hashSU) public {  
        SUCount++;  
        registry[SUCount] = hashSU;  
    }  
}
```

Codice dello **Smart Contract** che gestisce il salvataggio su blockchain delle **single SU**.



# Le operazioni disponibili

Condividere  
informazioni  
in modo  
sicuro  
combinando  
Git e  
Blockchain

Laureando:  
Paolo Speziali  
Relatore:  
Luca Grilli

Il Problema

L'Obiettivo

Il Software  
PineSU

Tecnologie  
utilizzate

Sviluppi futuri

- 1 **Creazione** di una Storage Unit o **Ricalcolo** di una Storage Unit pre-esistente.
- 2 **Staging** di una Storage Unit nello Storage Group.
- 3 **Registrazione** dello Storage Group nella Blockchain.
- 4 **Chiusura** di una Storage Unit.
- 5 **Esportazione** di sottoinsiemi di file da una Storage Unit.
- 6 **Controllo** di integrità di **singoli file** esportati da altre Storage Unit.
- 7 **Controllo** di integrità su una **Storage Unit**.



# Creazione di una Storage Unit (1 di 2)

Condividere  
informazioni  
in modo  
sicuro  
combinando  
Git e  
Blockchain

Laureando:  
Paolo Speciali  
Relatore:  
Luca Grilli

Il Problema

L'Obiettivo

Il Software  
PineSU

Tecnologie  
utilizzate

Sviluppi futuri

```
MINGW64/d:/Progetti/Tesi

PineSU

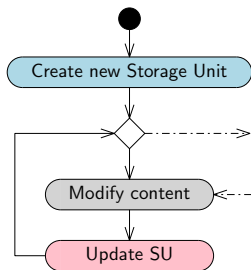
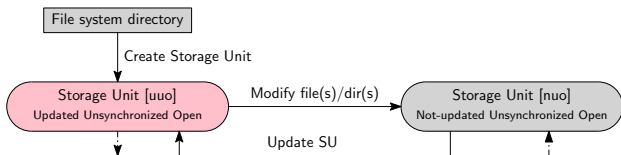
? Welcome to PineSU, choose the operation to perform (Use arrow keys)
> Create new SU / Recalculate open SU
  Stage Storage Unit for Synchronization
  Close current SU
  Register Staged SUs in the blockchain network
  Check SU integrity
  Export files from current SU
  Check files integrity
(Move up and down to reveal more choices)
```



# Creazione di una Storage Unit (2 di 2)

Condividere  
informazioni  
in modo  
sicuro  
combinando  
Git e  
Blockchain

Laureando:  
Paolo Speziali  
Relatore:  
Luca Grilli



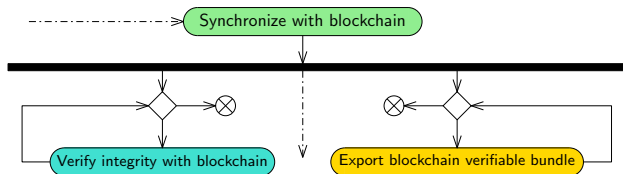
```
var merkleroot =  
  gitLogic.calculateTree(filelist);  
await inquirer.askSUDetails(  
  files.getCurrentDirectoryBase(),  
  remote).then((details) => {  
    details.owner = w1  
    details.hash = merkleroot  
    details.filelist = filelist  
    details.closed = false  
    files.saveJSON(details);  
  });
```



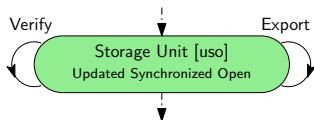
# Registrazione di uno Storage Group

Condividere  
informazioni  
in modo  
sicuro  
combinando  
Git e  
Blockchain

Laureando:  
Paolo Speziali  
Relatore:  
Luca Grilli



```
var [doc, openRoot, closedRoot] =
  files.createSGTrees(sg);
ethLogic.
  addToTree(openRoot, mc, false);
ethLogic.
  addToTree(closedRoot, mc, true);
var [oHash, cHash, transHash] =
  await ethLogic.registerMC(mc);
for(var el of document){
  el.transHash = transHash;
  files.createReg(el);
}
```







# Visualizzazione post-registrazione

Condividere  
informazioni  
in modo  
sicuro  
combinando  
Git e  
Blockchain

Laureando:  
Paolo Speziali  
Relatore:  
Luca Grilli

Il Problema

L'Obiettivo

Il Software  
PineSU

Tecnologie  
utilizzate

Sviluppi futuri

BLOCK 94	MINED ON 2021-08-18 16:00:20	GAS USED 21320
BLOCK 93	MINED ON 2021-08-18 15:27:51	GAS USED 21320
BLOCK 92	MINED ON 2021-08-18 15:26:23	GAS USED 21320
BLOCK 91	MINED ON 2021-08-18 15:25:55	GAS USED 21320

VALUE 0.00 ETH	GAS USED 21320
TX DATA 0xe67006f15ecd3fa2719d148be68d3a3242e1be8b	

## EVENTS

← BACK		BLOCK 94	
GAS USED 21320	GAS LIMIT 5	MINED ON 2021-08-18 16:00:20	BLOCK HASH 0xbaa08b6b640accbb9648ff313929998ced4a1abe14d6769756c
TX HASH 0xc563030328e652d427cd00707d7a0e2ce0bcf6c76b23482469eb497e2dc87d2e			
FROM ADDRESS 0xCF23544bFC002905532D086bF647754A84	TO CONTRACT ADDRESS 0x3a6990caE86a35a4022105b4c090DEF6490		VALUE 0
732966	8A0629		21320

```

"path":
  "D:/sample",
"root":
  "e67006f15ecd3[...]e6
  8d3a3242e1be8b",
"transHash":
  "0xc563030328e[...]69
  eb497e2dc87d2e"

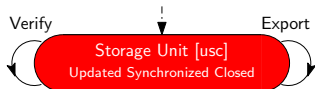
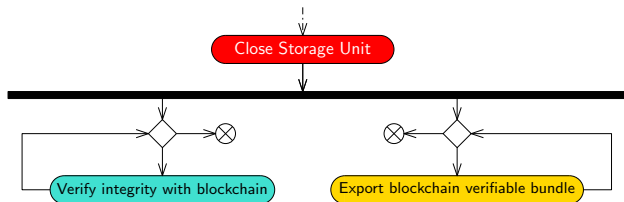
```



# Chiusura di una Storage Unit (2 di 2)

Condividere  
informazioni  
in modo  
sicuro  
combinando  
Git e  
Blockchain

Laureando:  
Paolo Speziali  
Relatore:  
Luca Grilli



```
if(fs.existsSync(".pinesu.json")){  
  var data =  
    fs.readFileSync(".pinesu.json")  
  var myObj = JSON.parse(data);  
  myObj.closed = true;  
  fs.writeFileSync(".pinesu.json",  
    JSON.stringify(myObj));  
  return myObj;  
}
```



# Esportazione di sottoinsiemi di Storage Unit

Condividere  
informazioni  
in modo  
sicuro  
combinando  
Git e  
Blockchain

Laureando:  
Paolo Speziali  
Relatore:  
Luca Grilli

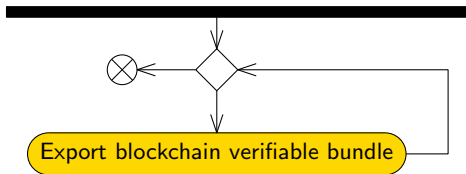
Il Problema

L'Obiettivo

Il Software  
PineSU

Tecnologie  
utilizzate

Sviluppi futuri



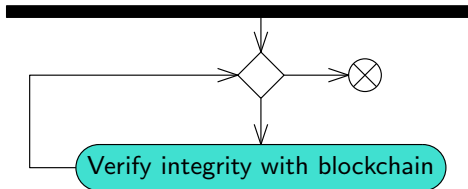
```
var zip = new AdmZip()
var fl = JSON.stringify(json)
zip.addFile(".pifiles.json", Buffer.alloc(fl.length, fl))
for(var el of list){
    zip.addLocalFile(path)
}
zip.writeZip("../pinesuExport.zip")
```



# Controllo d'integrità su una Storage Unit

Condividere  
informazioni  
in modo  
sicuro  
combinando  
Git e  
Blockchain

Laureando:  
Paolo Speciali  
Relatore:  
Luca Grilli



Il Problema

L'Obiettivo

Il Software  
PineSU

Tecnologie  
utilizzate

Sviluppi futuri

```
async verifyHash(transHash, hash){
  const res =
    await this.web3.eth.getTransaction(transHash)
  if(res.input == "0x"+hash){
    return true;
  } else {
    return false;
  }
}
```



# Node.js

Condividere  
informazioni  
in modo  
sicuro  
combinando  
Git e  
Blockchain

Laureando:  
Paolo Speciali  
Relatore:  
Luca Grilli

Il Problema

L'Obiettivo

Il Software  
PineSU

Tecnologie  
utilizzate

Sviluppi futuri

**Node.js** è un **ambiente di run-time**, che permette di eseguire codice **Javascript**.

Esso ha come obiettivi chiave l'**efficienza** e la **scalabilità**, può infatti eseguire velocemente codice Javascript sia **server-side** che **client-side**.

Parte fondamentale di Node sono i suoi numerosi **moduli**: librerie e framework realizzati dalla comunità e installabili con facilità tramite il package manager **npm**





# Moduli dei connettori

Condividere  
informazioni  
in modo  
sicuro  
combinando  
Git e  
Blockchain

Laureando:  
Paolo Speciali  
Relatore:  
Luca Grilli

Il Problema

L'Obiettivo

Il Software  
PineSU

Tecnologie  
utilizzate

Sviluppi futuri

**web3.js** è un **modulo npm** che permette di interagire con **node Ethereum** locali e remoti.

*PineSU EC* lo utilizza per effettuare le **transazioni** con i suoi wallet e per comunicare con lo **Smart Contract**.



**Simple Git** è un **modulo npm** che permette di comunicare con il **client Git** locale.

Usato in *PineSU GC*, esso permette l'esecuzione di comandi in maniera **asincrona**.



## Altri moduli

Condividere  
informazioni  
in modo  
sicuro  
combinando  
Git e  
Blockchain

Laureando:  
Paolo Speciali  
Relatore:  
Luca Grilli

Il Problema

L'Obiettivo

Il Software  
PineSU

Tecnologie  
utilizzate

Sviluppi futuri



**Inquirer.js** è un **modulo npm** che facilita la creazione di **interfacce utente** tramite menù testuali.

In *PineSU CLI* viene usato per **interagire** con l'utente ponendogli **domande** dalla risposta chiusa o aperta.

**ADM-ZIP** è un **modulo npm** che consente di creare **cartelle compresse** in formato ZIP.

*PineSU BEL* lo utilizza per **esportare sottoinsiemi** di SU mantenendo la **struttura gerarchica** originale.





# Sviluppi futuri

Condividere  
informazioni  
in modo  
sicuro  
combinando  
Git e  
Blockchain

Laureando:  
Paolo Speciali  
Relatore:  
Luca Grilli

Il Problema

L'Obiettivo

Il Software  
PineSU

Tecnologie  
utilizzate

Sviluppi futuri

- 1 Migliorare gestione degli **accumulatori crittografici** per le **single SU**.
- 2 Impedire tramite **Smart Contract** la modifica di SU chiuse.
- 3 Aggiungere **connettori** per **ulteriori blockchain**.
- 4 Creare **portale web** con **server Git** per la gestione remota delle SU.





Condividere  
informazioni  
in modo  
sicuro  
combinando  
Git e  
Blockchain

Laureando:  
Paolo Speziali  
Relatore:  
Luca Grilli

Il Problema

L'Obiettivo

Il Software  
PineSU

Tecnologie  
utilizzate

Sviluppi futuri



# Gli strumenti attuali per la condivisione di documenti

Condividere  
informazioni  
in modo  
sicuro  
combinando  
Git e  
Blockchain

Laureando:  
Paolo Speziali  
Relatore:  
Luca Grilli

Il Problema

L'Obiettivo

Il Software  
PineSU

Tecnologie  
utilizzate

Sviluppi futuri

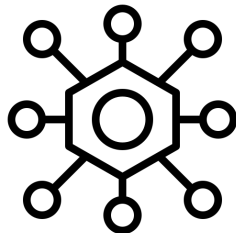
Uno strumento digitale solitamente segue uno di questi due paradigmi:

**centralizzato** e **distribuito**.

Nel primo un'entità centrale si occupa dell'**immagazzinamento** e della **verifica** dei dati degli utenti.

Ciò ha diversi **svantaggi**:

- Potenziali attacchi all'entità
- Possibile uso malevolo dei nostri dati
- Alti costi d'intermediazione





# Strumenti distribuiti

Condividere  
informazioni  
in modo  
sicuro  
combinando  
Git e  
Blockchain

Laureando:  
Paolo Speciali  
Relatore:  
Luca Grilli

Il Problema

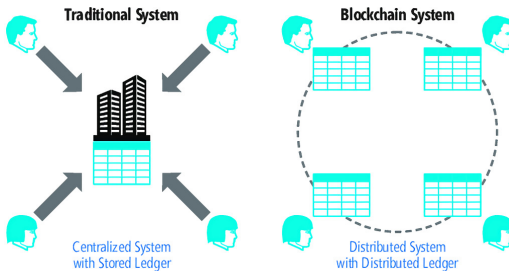
L'Obiettivo

Il Software  
PineSU

Tecnologie  
utilizzate

Sviluppi futuri

Usando invece **un'architettura distribuita**, sia per la **gestione dei file**, sia per la **verifica delle informazioni**, saremo in grado costruire uno strumento che può affidarsi alla parola di una **moltitudine di entità**, rendendo molto più complicati e rilevabili attacchi e manomissioni.





# Accumulatori crittografici

Condividere  
informazioni  
in modo  
sicuro  
combinando  
Git e  
Blockchain

Laureando:  
Paolo Speciali  
Relatore:  
Luca Grilli

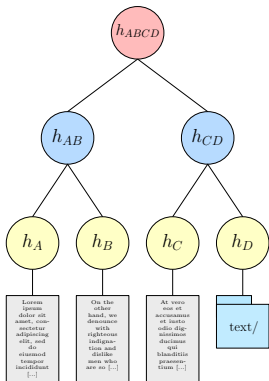
Il Problema

L'Obiettivo

Il Software  
PineSU

Tecnologie  
utilizzate

Sviluppi futuri



Strumenti che **comprimono molte informazioni** in una **costante** di dimensione fissa.

Un esempio ne sono i **Merkle Tree**, alberi binari in cui ogni foglia corrisponde all'hash di un elemento.

Risalendo ogni nodo interno calcolerà il proprio hash con gli hash dei nodi figli, l'hash della root sarà **univoco** a quelle foglie in quell'ordine.



# Gli accumulatori di PineSU

Condividere  
informazioni  
in modo  
sicuro  
combinando  
Git e  
Blockchain

Laureando:  
Paolo Speciali  
Relatore:  
Luca Grilli

Il Problema

L'Obiettivo

Il Software  
PineSU

Tecnologie  
utilizzate

Sviluppi futuri

- *SU Merkle Tree*: Le sue **foglie** sono gli **hash dei file e directory** della SU. La sua **root** è l'**hash della SU** stessa.
- *Storage Group (SG)*: Le sue **foglie** sono le **SU da registrare** su blockchain nella prossima transazione.
- *Merkle Calendar (MC)*.

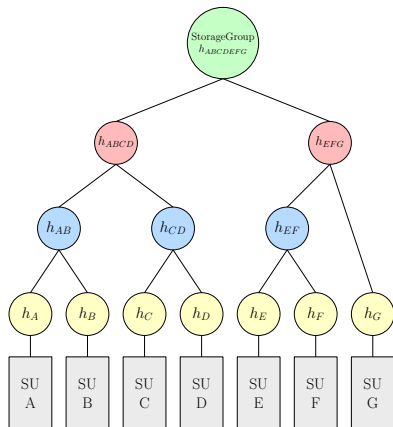


Figura: Uno Storage Group