Condividere informazioni in modo sicuro combinando Git e Blockchain

Studente: Paolo Speziali
Relatore: Luca Grilli

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

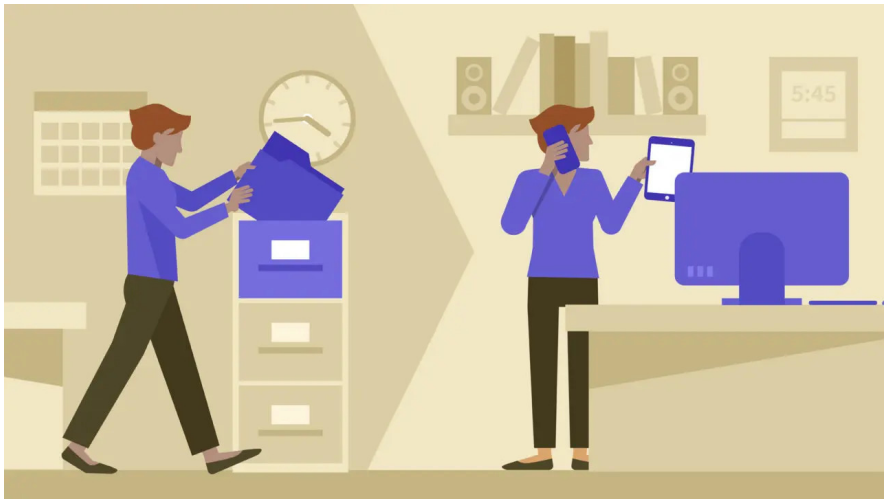
A.D. 1308
unipg
DIPARTIMENTO
DI INGEGNERIA

A.A. 2020/2021

Indice

La digitalizzazione

È in atto, negli ultimi anni, un piano di **digitalizzazione** delle PA. Esso mira all'evoluzione tecnologica di tutte le sue mansioni e alla creazione di portali web per il cittadino.
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**.



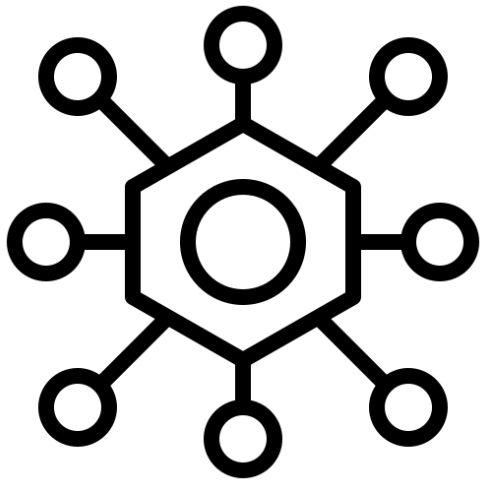
Il problema della burocrazia

Il più grande avversario della digitalizzazione è la **burocrazia** italiana: i suoi processi sono **lenti** e **complessi** anche a causa dell'**importanza** dei documenti da gestire. È necessaria una **sburocratizzazione** grazie a degli strumenti digitali che permettano di **salvare**, **validare** e **condividere** documenti senza abbassare il **livello di sicurezza**.

Gli strumenti attuali

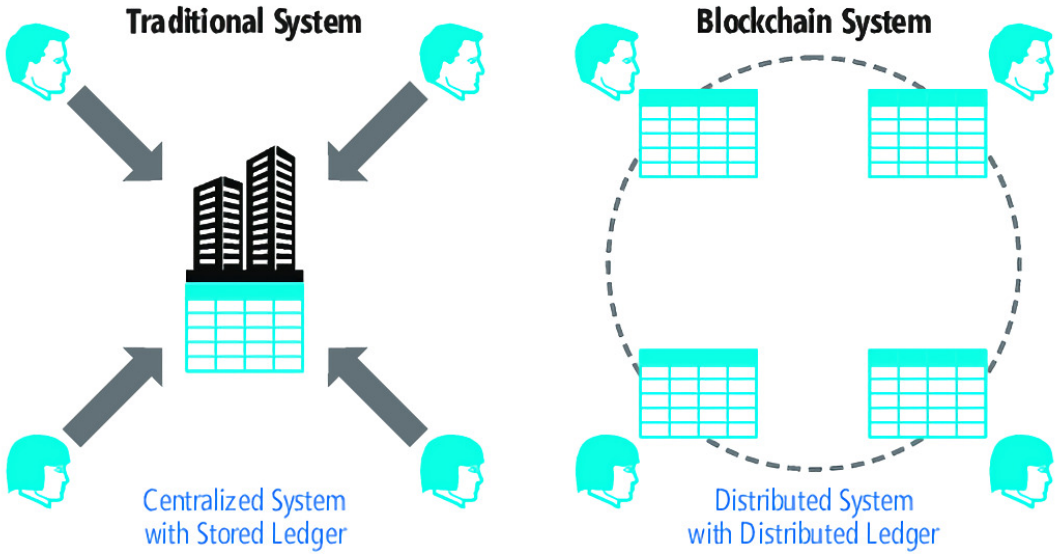
Uno strumento digitale solitamente segue uno di questi due paradigmi: **centralizzato** e **distribuito**.
Nel primo un'entità centrale si occupa dell'**immagazzinamento** e della **verifica** dei dati degli utenti.
Ciò ha diversi **svantaggi**:

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



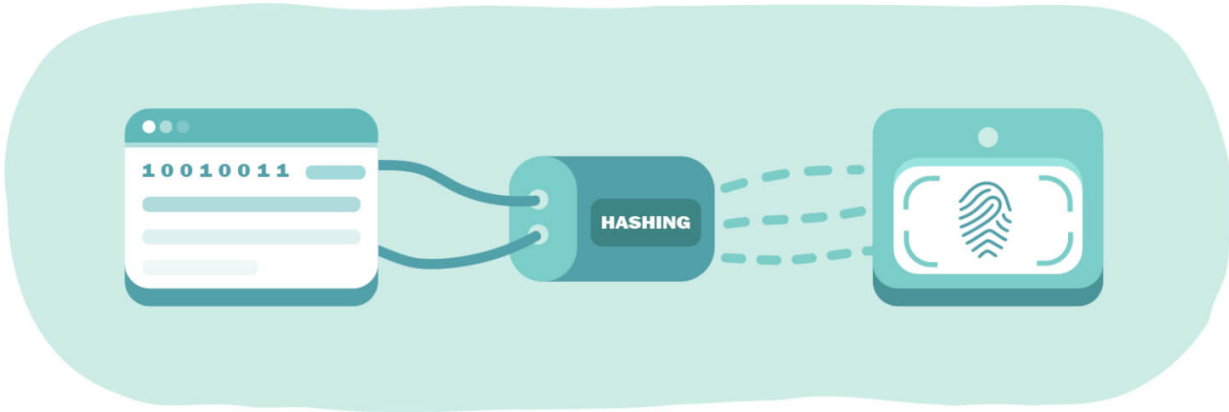
Strumenti distribuiti

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.



Funzioni crittografiche di hashing

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**.



Git

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. . .

Blockchain

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

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**.

Perché blockchain?

L'utilizzo della blockchain nel progetto è giustificato da:

- Immutabilità $\rightarrow$ Garantisce integrità dei dati

[1] Single Point Of Failure

Perché blockchain?

L'utilizzo della blockchain nel progetto è giustificato da:

- Immutabilità $\rightarrow$ Garantisce integrità dei dati
- Decentralizzazione $\rightarrow$ Resistenza allo **SPOF**[1]

[1] Single Point Of Failure

Perché blockchain?

L'utilizzo della blockchain nel progetto è giustificato da:

- Immutabilità → Garantisce integrità dei dati
- Decentralizzazione → Resistenza allo **SPOF**[1]
- Disintermediazione → Eliminazione di *middle-men* e dei loro costi

[1] Single Point Of Failure

Perché blockchain?

L'utilizzo della blockchain nel progetto è giustificato da:

- Immutabilità → Garantisce integrità dei dati
- Decentralizzazione → Resistenza allo **SPOF**[1]
- Disintermediazione → Eliminazione di *middle-men* e dei loro costi
- Validazione *peer-to-peer* → Potere distribuito

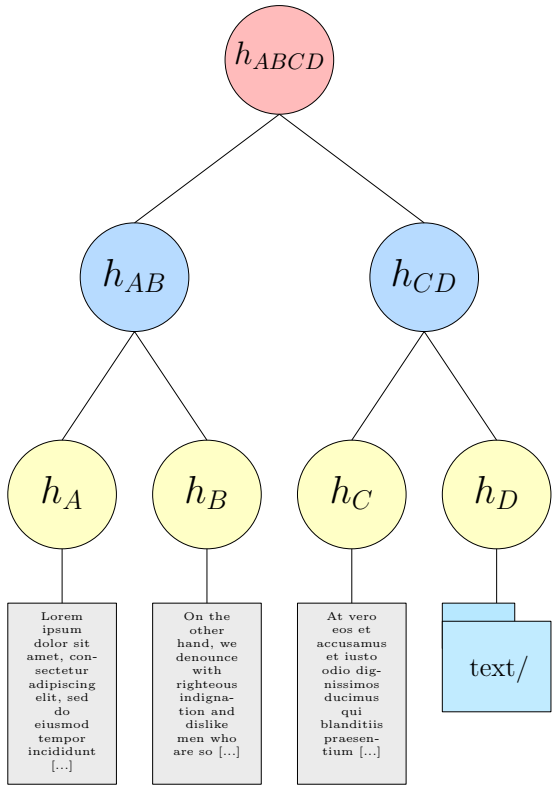
[1] Single Point Of Failure

Il problema della blockchain

Vogliamo usare la blockchain per **immagazzinare** informazioni, ciò è problematico: **più dati** vorremo registrare, **più dovremo pagare**. Occorre trovare una soluzione per registrare **pochi dati** ma utilizzabili per **numerosi controlli** in **breve tempo**. La soluzione è l'utilizzo di **accumulatori crittografici**.

Accumulatori crittografici

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.



L'Obiettivo

Realizzare uno **strumento digitale distribuito** in grado, tramite interazioni con **Git** e la **blockchain**, di:

Salvare hash di repository su blockchain

Esportare sottoinsiemi di repository verificabili

Verificare l'integrità di singoli file e repository

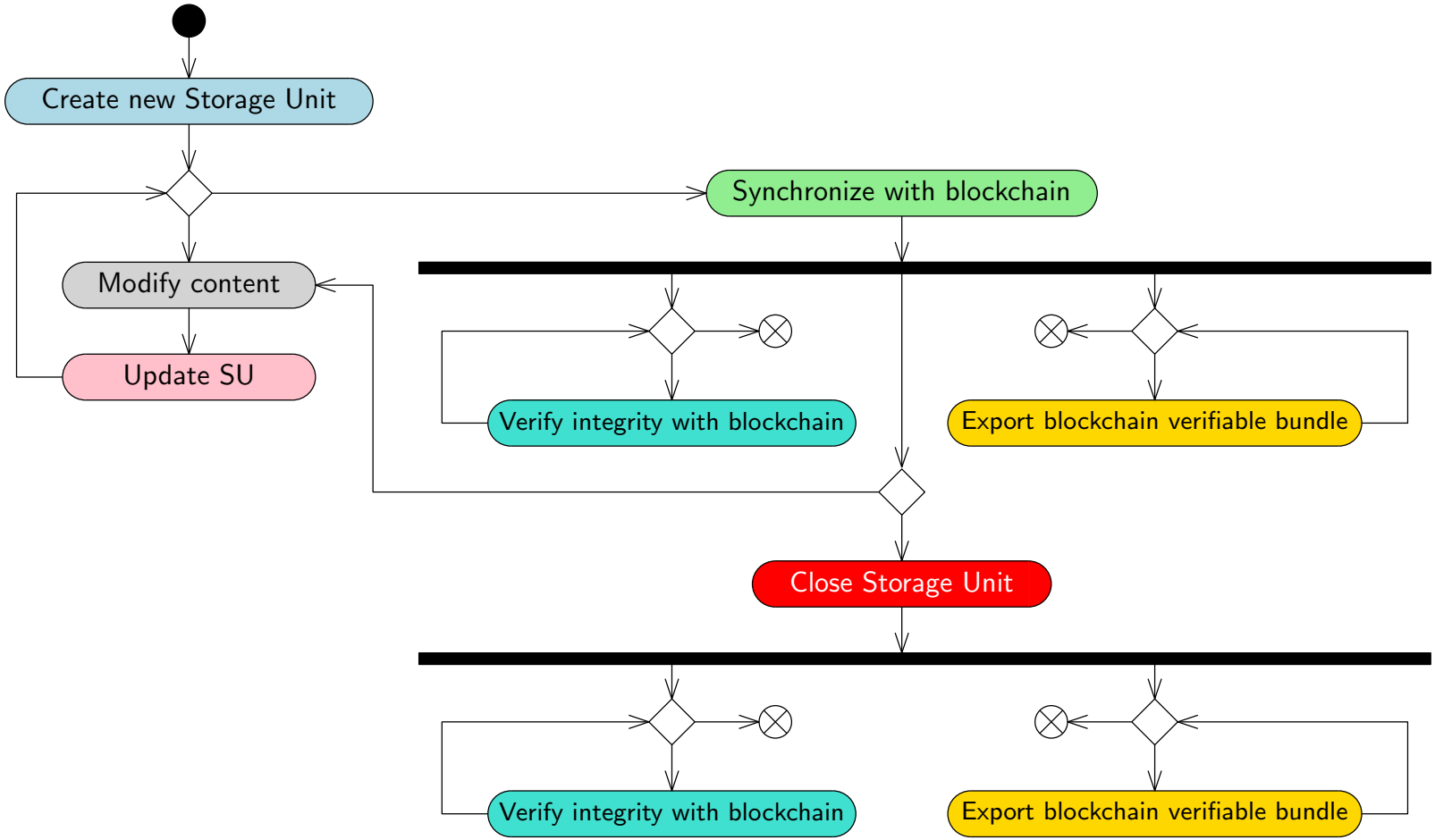
Il Software PineSU

PineSU è un software **Javascript** che sfrutta il runtime **Node.js**.

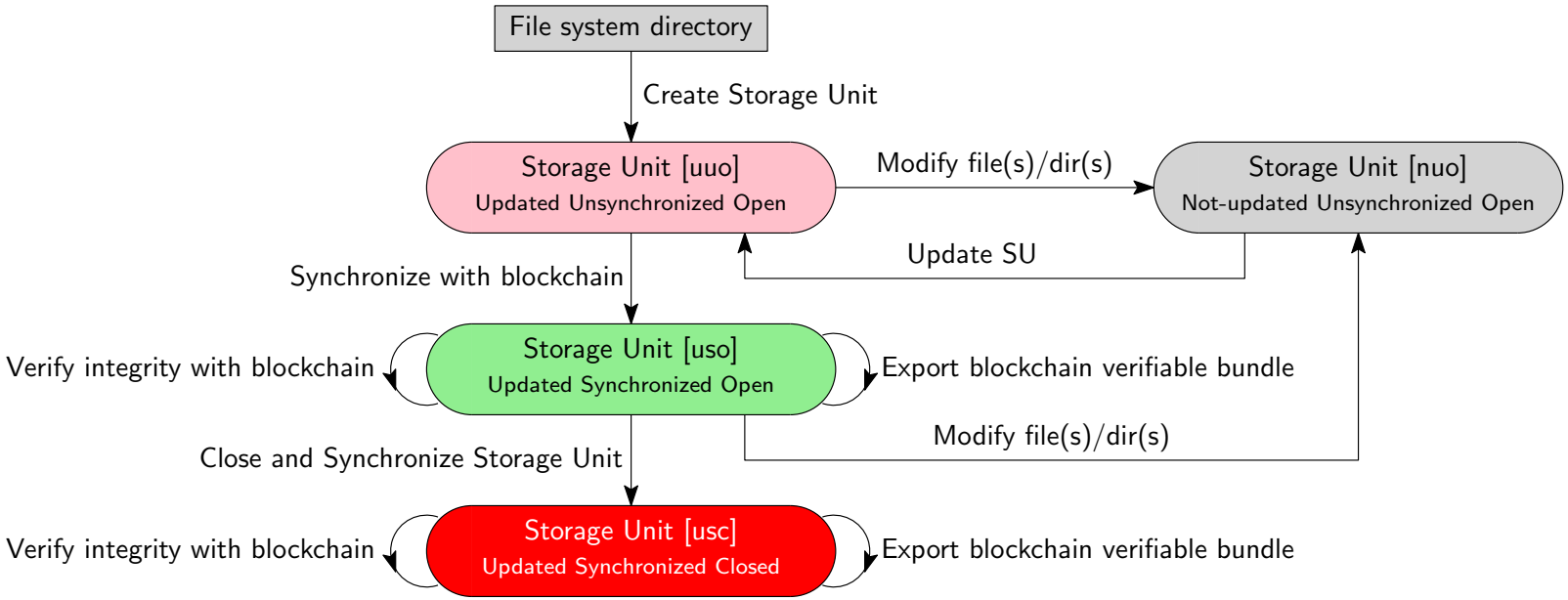
L'applicazione crea delle **strutture** sulle repository Git chiamate **Storage Unit** (SU) tramite metadati.

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.

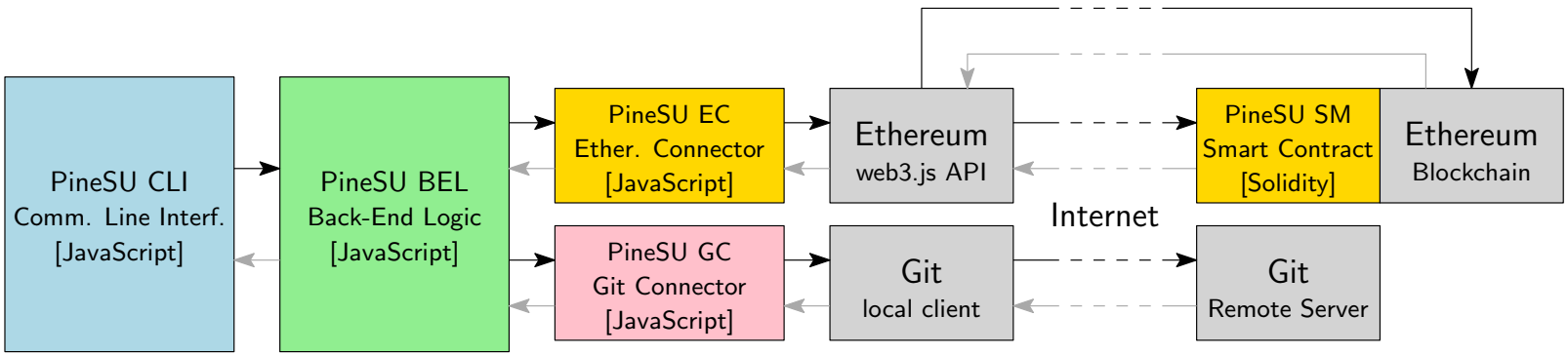
Workflow



Ciclo vitale di una Storage Unit



Architettura



Architettura (Cont.)

- *PineSU* **CLI** *(Command Line Interface)*: Modulo che crea l'interfaccia utente con cui è possibile interagire e richiama le funzioni degli altri moduli di conseguenza.
- *PineSU* **BEL** *(Back End Logic)*: Questo componente è il nucleo di PineSU. Gestisce tutte 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.

Gli accumulatori di PineSU

- *SU Merkle Tree*: Le sue foglie sono gli hash dei file e directory della SU, la sua root è l'hash della SU stessa.

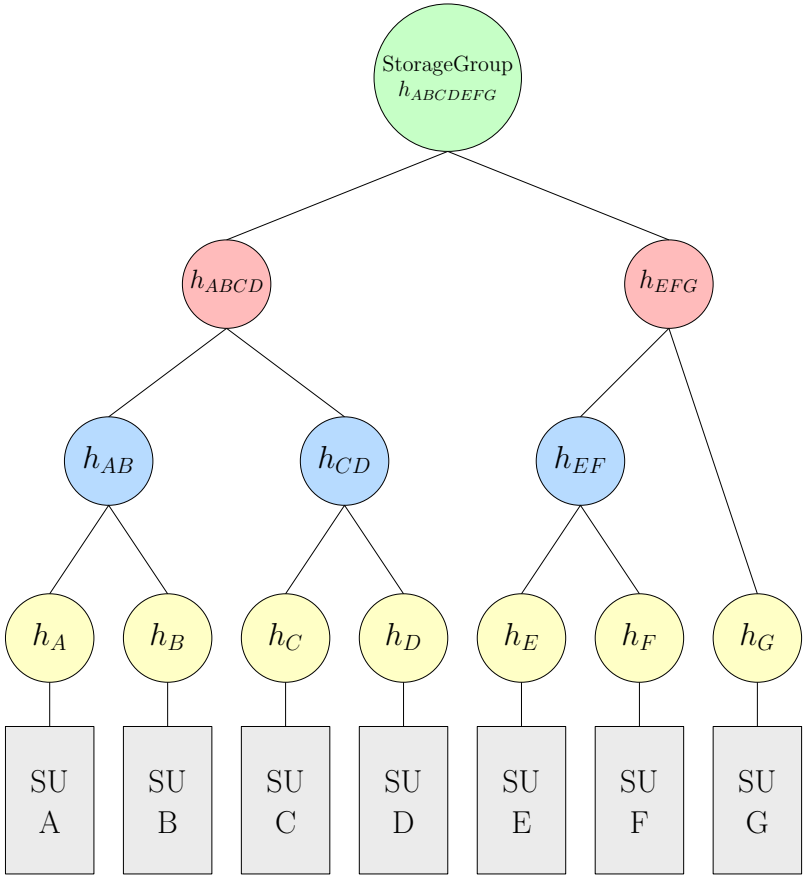


Figura: Uno Storage Group

Gli accumulatori di PineSU

- *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.

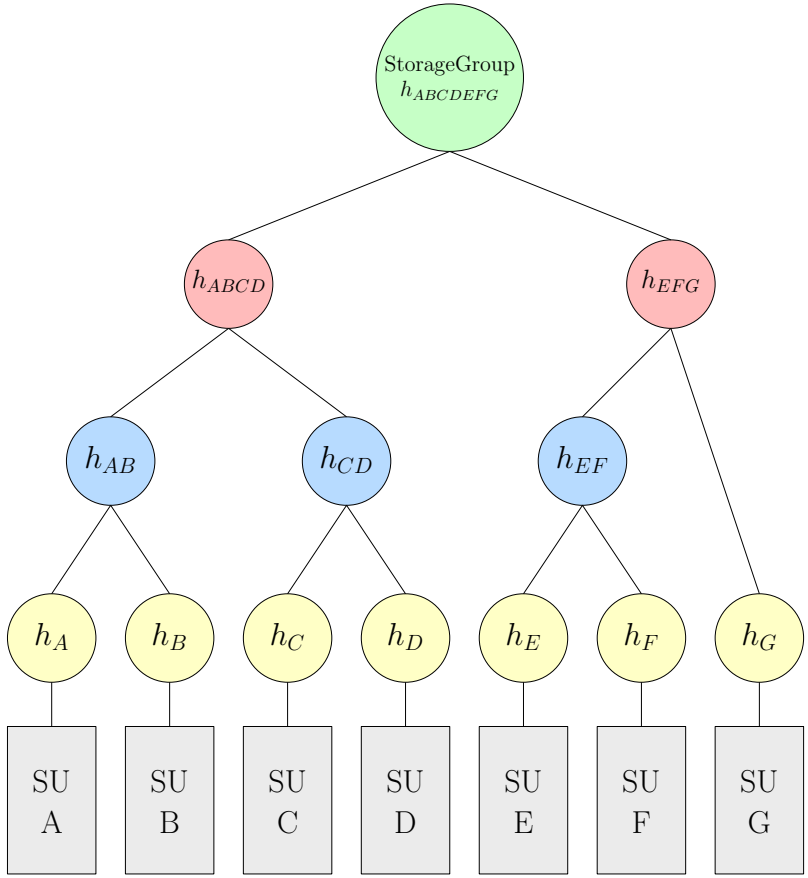


Figura: Uno Storage Group

Gli accumulatori di PineSU

- *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***)*.

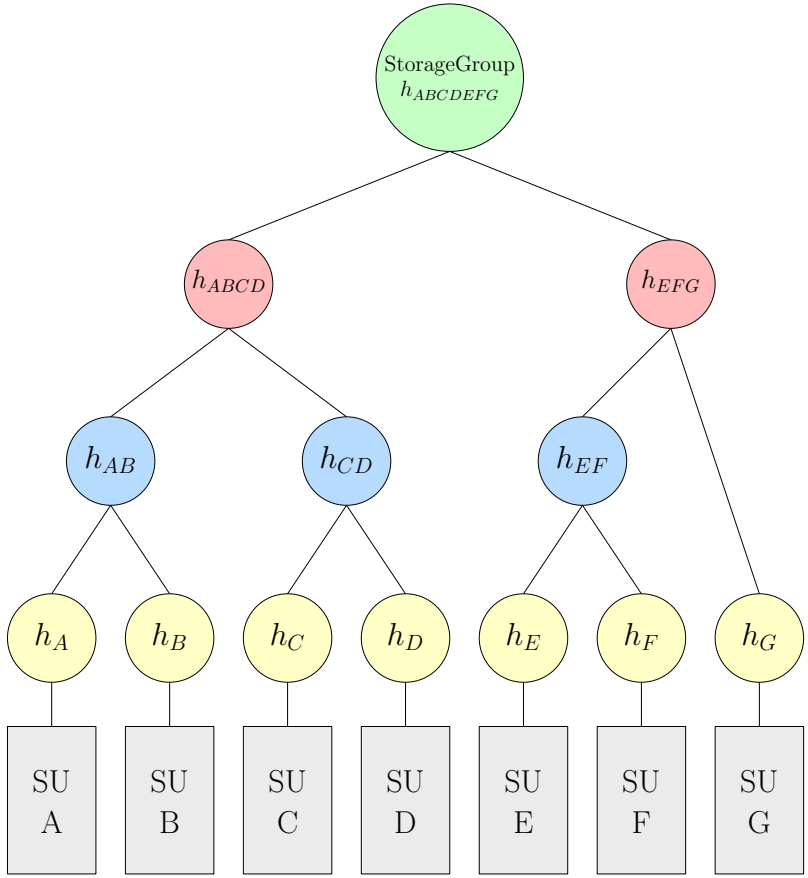


Figura: Uno Storage Group

Gli accumulatori di PineSU (Cont.)

Un **Merkle Calendar** è 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.

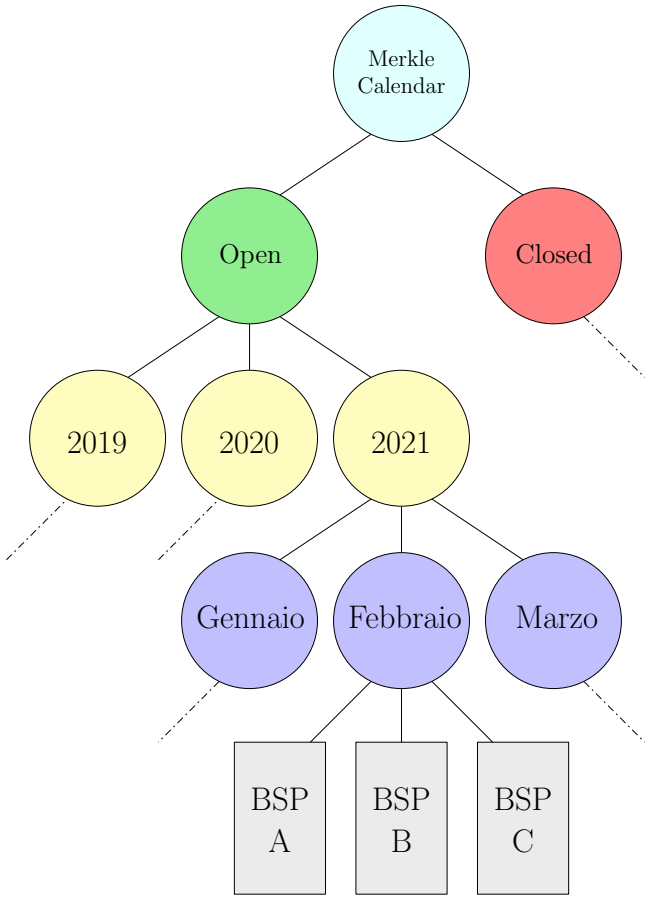


Figura: Un Merkle Calendar

Le operazioni disponibili

1. Creazione di una Storage Unit o Ricalcolo di una Storage Unit pre-esistente

Le operazioni disponibili

1. Creazione di una Storage Unit o Ricalcolo di una Storage Unit pre-esistente
2. Staging di una Storage Unit nello Storage Group

Le operazioni disponibili

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

Le operazioni disponibili

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

Le operazioni disponibili

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 SU

Le operazioni disponibili

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 SU
6. Controllo di integrità di singoli file esportati da altre SU

Le operazioni disponibili

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 SU
6. Controllo di integrità di singoli file esportati da altre SU
7. Controllo di integrità su una SU