

**PROGETTO CYBERSECURITY: BLOCKCHAIN**

Corso di CyberSecurity

**GRUPPO 10**

Angelo Di Carlo

Alessandro Sebastianelli

Giulia Morgoni

Lorenzo Medici

Mirko Simoni

Prof. Luca Spalazzi

Anno Accademico 2019/2020

Sommario

[INTRODUZIONE 3](#_Toc67057818)

[RACCOLTA ED ANALISI DEI REQUISITI 3](#_Toc67057819)

[EARLY REQUIREMENT ANALYSIS 4](#_Toc67057820)

[LATE REQUIREMENT ANALYSIS 5](#_Toc67057821)

[IDENTIFICAZIONE ASSET 6](#_Toc67057822)

[USE CASE SPECIFICATION 6](#_Toc67057823)

[ASSET EVALUATION AND EXPOSURE 9](#_Toc67057824)

[IDENTIFICAZIONE MINACCE 10](#_Toc67057825)

[ABUSE E MISUSE CASE 10](#_Toc67057826)

[ATTACK TREE 18](#_Toc67057827)

[ANALISI DEL RISCHIO 22](#_Toc67057828)

[ANALISI TECNICHE DI MITIGAZIONE 23](#_Toc67057829)

[IMPLEMENTAZIONE 28](#_Toc67057830)

[TECNOLOGIE PER L’IMPLEMENTAZIONE 28](#_Toc67057831)

[ETHEREUM e QUORUM 28](#_Toc67057832)

[SOLIDITY e JAVASCRIPT 29](#_Toc67057833)

[NODE.JS e EXPRESS 29](#_Toc67057834)

[GANACHE 30](#_Toc67057835)

[REACT 30](#_Toc67057836)

[MONGODB 30](#_Toc67057837)

[IPFS (InterPlanetary File System) 30](#_Toc67057838)

[DETTAGLI IMPLEMENTATIVI 31](#_Toc67057839)

[SVILUPPO APPLICAZIONE 31](#_Toc67057840)

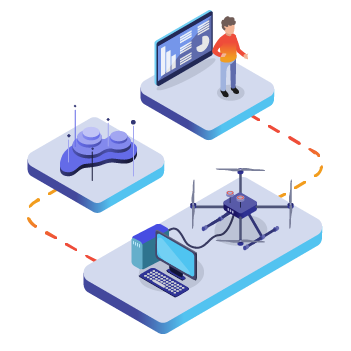
[SVILUPPO BLOCKCHAIN E SMART CONTRACT 34](#_Toc67057841)

[IMPLEMENTAZIONE GANACHE 39](#_Toc67057842)

[MANUALE WEB APP 40](#_Toc67057843)

[PROGRESSI FUTURI 42](#_Toc67057844)

# INTRODUZIONE

Il progetto da noi realizzato consiste nell’implementazione di un sistema informatizzato che permetta di rendere sicuro e monitorato un ambiente edilizio, evitando comportamenti illeciti.

Il lavoro edilizio di cui ci occupiamo è rappresentabile dai seguenti attori:

* DRONE: dispositivo che permette di acquisire immagini relative al territorio in cui si costruisce
* IMPRESA: impresa vincitrice di una gara appaltante
* SERVIZIO DI FOTOGRAMMETRIA: software esterno che si occupa di calcolare le misure relative alle immagini provenienti dal drone.

Le attività svolte dai tre attori sono collegate tra loro secondo una certa sequenzialità: il drone cattura le immagini che vengono memorizzate in un database interno all’azienda per poi essere analizzate in modo da selezionare le migliori, sia qualitativamente sia per la loro rilevanza; queste scelte verranno poi inviate al servizio di fotogrammetria, che si occuperà di estrarne le misure. Queste misure verranno poi registrate all’interno del giornale dei lavori, documento nel quale vengono memorizzate tutte le attività, passo dopo passo, svolte dall’impresa.

Quest’ultimo è il documento più importante nel quale vengono registrati gli avvenimenti quotidiani rilevanti che potrebbero influire sui lavori e alterarne il normale andamento. Esso viene gestito dal direttore dei lavori, un ingegnere esterno all’impresa esecutrice nominato dall’ente appaltante per controllare l’operato dell’azienda edile a cui è stato assegnato l’appalto, evitando così comportamenti irregolari.

Il nostro obiettivo è quindi quello di garantire una maggiore sicurezza specialmente nelle fasi di store delle immagini, ottenimento delle misure e nella fase di trascrizione della documentazione del giornale dei lavori.

Ciò è implementabile grazie all’utilizzo delle blockchain, una struttura dati condivisa ed immutabile che garantisce l’integrità delle immagini, l’autenticità e la sicurezza dei dati sfruttando la crittografia.

# RACCOLTA ED ANALISI DEI REQUISITI

Nella fase iniziale del progetto risulta necessario analizzare le informazioni a disposizione che ci permetteranno di implementare un progetto che rispetti le specifiche richieste.

I requisiti preliminari sono i seguenti:

* il drone verrà utilizzato per scattare delle immagini le quali, dopo essere registrate in un database e selezionate, vengono assegnate delle misurazioni tecniche grazie ad un servizio di fotogrammetria;
* le misurazioni vengono registrate all’interno del giornale dei lavori per poterne verificare l’integrità, l’autenticità e la veridicità.

Gli attori principali che operano all’interno del nostro cantiere sono essenzialmente 3: il drone, il direttore dei lavori ed il servizio di fotogrammetria.

## EARLY REQUIREMENT ANALYSIS

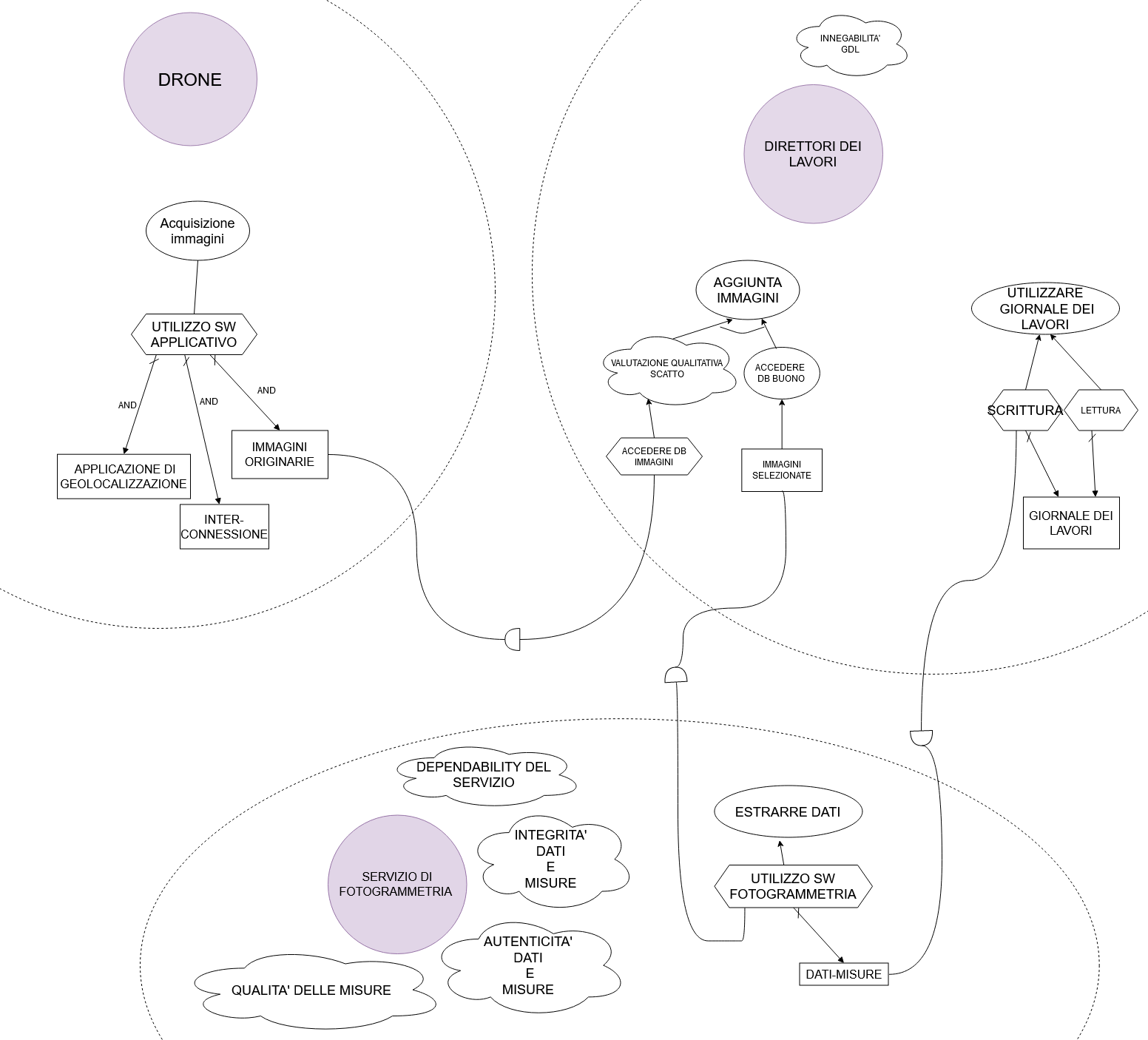
Per rendere il tutto più chiaro e comprensibile è stato possibile implementare uno schema di base, utilizzando il linguaggio di modellazione i\*, nel quale vengono rappresentati gli attori, i collegamenti tra loro e le mansioni che potevano svolgere.

Figura 1

Analizziamo gli attori:

* *Drone*: si occupa di catturare le immagini del cantiere, per poi memorizzarle all’interno di un database di appoggio che verrà poi scremato e dal quale verranno estratte solo le immagini qualitativamente migliori. Inoltre, ad ogni immagine, vengono associati due attributi: uno riguardante la geo localizzazione, e un altro riguardante l’interconnessione tra le immagini stesse.
* *Direttore dei lavori*: si occupa di selezionare le immagini con il quale lavorare inserendole successivamente in un nuovo database e di gestire il giornale dei lavori, controllandone le operazioni di scrittura e di lettura.
* *Servizio di fotogrammetria*: si occupa di estrarre misure specifiche partendo dalle immagini presenti nel database delle immagini selezionate e di inviare questi dati al direttore dei lavori che si impegnerà a trascriverli nel giornale dei lavori.

## LATE REQUIREMENT ANALYSIS

Il diagramma riportato sopra non tiene conto del sistema software, componente principale di un sistema di cybersecurity. In questa fase di progettazione andremo ad interporre il sistema software tra i vari attori, ridirezionando le loro dipendenze.

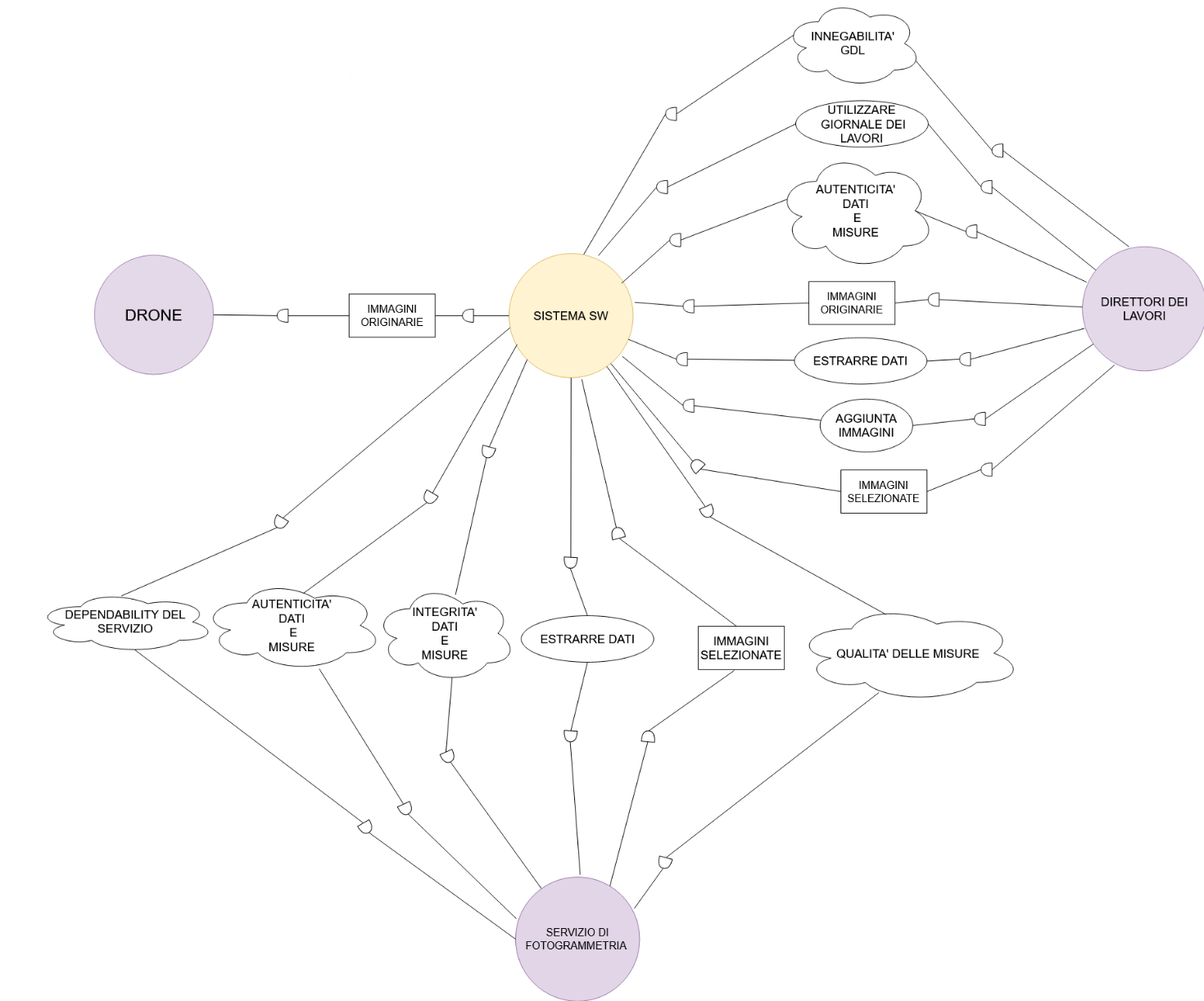


Figura 2

Nel nostro caso abbiamo connesso i tre attori con il sistema software centrale tramite delle relazioni specifiche di dipendenza. I collegamenti sono esplicitati con dei costrutti messi a disposizione dal modello i\* che rappresentano dei goal, dei soft goal e delle risorse.

## IDENTIFICAZIONE ASSET

Per poter definire delle misure di sicurezza appropriate a tale sistema in grado di proteggerlo, occorre identificare gli asset fondamentali e più importanti per il progetto.

Nel nostro caso gli asset principali sono:

* Database delle immagini
* Giornale dei lavori
* Dati
* Utilizzo del servizio di fotogrammetria

### USE CASE SPECIFICATION

Per ogni asset è stato realizzato un Use Case Specification, ovvero una rappresentazione tabellare in grado di definire gli attori, i tipi di dato utilizzati ed i vari flussi interni al software. Ciò permette di focalizzarci sugli asset stessi in modo tale da poter individuare i possibili attacchi a cui potrebbero essere sottoposti e, di conseguenza, le relative azioni di mitigazione.

|  |  |
| --- | --- |
| Use Case ID: DM01 Use case Name: Dati | |
| Attori | Servizio di fotogrammetria, Direttore dei lavori |
| Descrizioni | Questi sono i dati estratti dal servizio di fotogrammetria, dalle immagini fornite dal drone |
| Asset | Dati |
| Pre-condizioni | L’immagine deve essere presente ed accessibile al software di fotogrammetria |
| Flusso standard | 1. Il drone scatta la foto 2. Il direttore dei lavori deve scegliere l’immagine 3. Il direttore dei lavori carica nel database la foto 4. Il software di fotogrammetria estrae i dati dalla foto |
| Eccezione di flusso | * Il software di fotogrammetria non riesce ad accedere alla foto * I dati non pervengono al direttore dei lavori |
| Post-condizioni | I dati vengono registrati sul giornale dei lavori |
| Requisiti non funzionali | Integrità, autenticità |

|  |  |
| --- | --- |
| Use Case ID: USF01 Use case Name: Servizio di fotogrammetria | |
| Attori | Servizio di fotogrammetria, |
| Descrizioni | Viene utilizzato per l’estrazione dei dati e delle misure |
| Asset | Database immagini, dati |
| Pre-condizioni | Il direttore dei lavori deve aver scelto l’immagine da far analizzare |
| Flusso standard | 1. Il direttore dei lavori sceglie e carica l’immagine nel database 2. Viene fornito l’accesso al software di fotogrammetria 3. Il software di fotogrammetria analizza l’immagine e ne estrae i dati 4. I dati sono passati al direttore dei lavori per aggiungerli al giornale dei lavori |
| Eccezione di flusso | * il software di fotogrammetria non riesce a leggere l’immagine * Il software di fotogrammetria riconosce che l’immagine è stata manomessa e non estrae i dati |
| Post-condizioni | I dati vengono estratti in modo corretto |
| Requisiti non funzionali | Affidabilità |

|  |  |
| --- | --- |
| Use Case ID: ADI01 Use case Name: Database immagini | |
| Attori | Direttore dei lavori, servizio di fotogrammetria |
| Descrizioni | Le immagini che per il direttore dei lavori sono utilizzabili per una futura estrazione dei dati, dal servizio di fotogrammetria, sono salvate nel database |
| Asset | Immagini provenienti dal drone |
| Pre-condizioni | Il drone deve aver scattato le foto |
| Flusso standard | 1. Il drone scatta la foto 2. Il drone salva la foto sul database 3. Il direttore dei lavori accede al database e seleziona le immagini da far elaborare al software di fotogrammetria 4. Il direttore dei lavori carica le immagini selezionate su un altro database |
| Eccezione di flusso | * La qualità delle immagini scattate è scadente |
| Post-condizioni | Salvataggio avvenuto correttamente |
| Requisiti non funzionali | Autenticità, non ripudio |

|  |  |
| --- | --- |
| Use Case ID: UGDL01 Use case Name: Giornale dei lavori | |
| Attori | Direttore dei lavori |
| Descrizioni | Il direttore dei lavori può accedere al giornale dei lavori per aggiornarlo o controllare i lavori |
| Asset | Dati, Giornale dei lavori |
| Pre-condizioni | * Per aggiungere delle misure, queste devono essere state estratte dal software di fotogrammetria * Per accedere in lettura, il giornale dei lavori deve avere contenuto informativo |
| Flusso standard | 1. Il software di fotogrammetria fornisce dati al direttore dei lavori 2. Il direttore dei lavori controlla questi dati prima di caricarli 3. In caso di dati coerenti li carica sul giornale dei lavori |
| Eccezione di flusso | Il direttore dei lavori non riesce ad accedere al giornale dei lavori |
| Post-condizioni | * Il direttore dei lavori vuole controllare il lavoro svolto durante l’ultimo periodo * Accede al giornale dei lavori e ottiene le informazioni di suo interesse |
| Requisiti non funzionali | Non ripudio, integrità, autenticità, affidabilità |

### ASSET EVALUATION AND EXPOSURE

Per effettuare una buona programmazione in grado di garantire la sicurezza degli asset fondamentali, è necessario assegnare loro due indici qualitativi, uno che per il valore, e uno per l’impatto se attaccato.

Gli indici numerici attribuiti al valore ed all’impatto sono stati decisi utilizzando come range di riferimento una scala di Likert a 7 valori.

|  |  |  |
| --- | --- | --- |
| **Asset** | **Valore** | **Impatto** |
| Dati | 5  La risorsa è fondamentale per il corretto completamento del giornale dei lavori. Usati per controllare il prodotto realizzato | 5 I dati devono essere coerenti e provenienti dall’elaborazione delle foto del drone. Una loro alterazione può portare a danni |
| Utilizzo del software di fotogrammetria | 4  Da questo task dipende direttamente la qualità e integrità dei dati. Valore medio. | 7  Un cattivo utilizzo del software può portare a un’errata estrazione di dati.  Valore alto. |
| Database immagini | 7  Poiché le immagini da elaborare vengono salvate nel database, assumono un alto valore. | 3  Dovuto al fatto che, grazie a delle tecniche di backup, è possibile riottenere i dati originali in casi di violazione. |
| Giornale dei lavori | 6  Permette il salvataggio delle informazioni riguardo il progetto e quindi è di fondamentale importanza. | 7  Una violazione della sicurezza potrebbe portare a informazioni non veritiere riguardo allo stato dell’avanzamento dei lavori che però verrebbero considerate attendibili.  Valore alto. |

## IDENTIFICAZIONE MINACCE

Successivamente all’identificazione degli aspetti principali del sistema, per ottenere un sistema sicuro si devono identificare ed analizzare le minacce, in modo da poter attivare in maniera preventiva le azioni di mitigazione.

### ABUSE E MISUSE CASE

Una considerazione da fare sui presunti attacchi è che questi possono essere effettuati da due tipologie completamente diverse di utenti:

* *Malevolo*: colui che effettua un attacco con lo scopo di rubare informazioni, di manomettere attività interne al sistema, ...
* *Sbadato*: colui che effettua un attacco senza alcuno scopo ma soltanto a causa di un errore di distrazione.

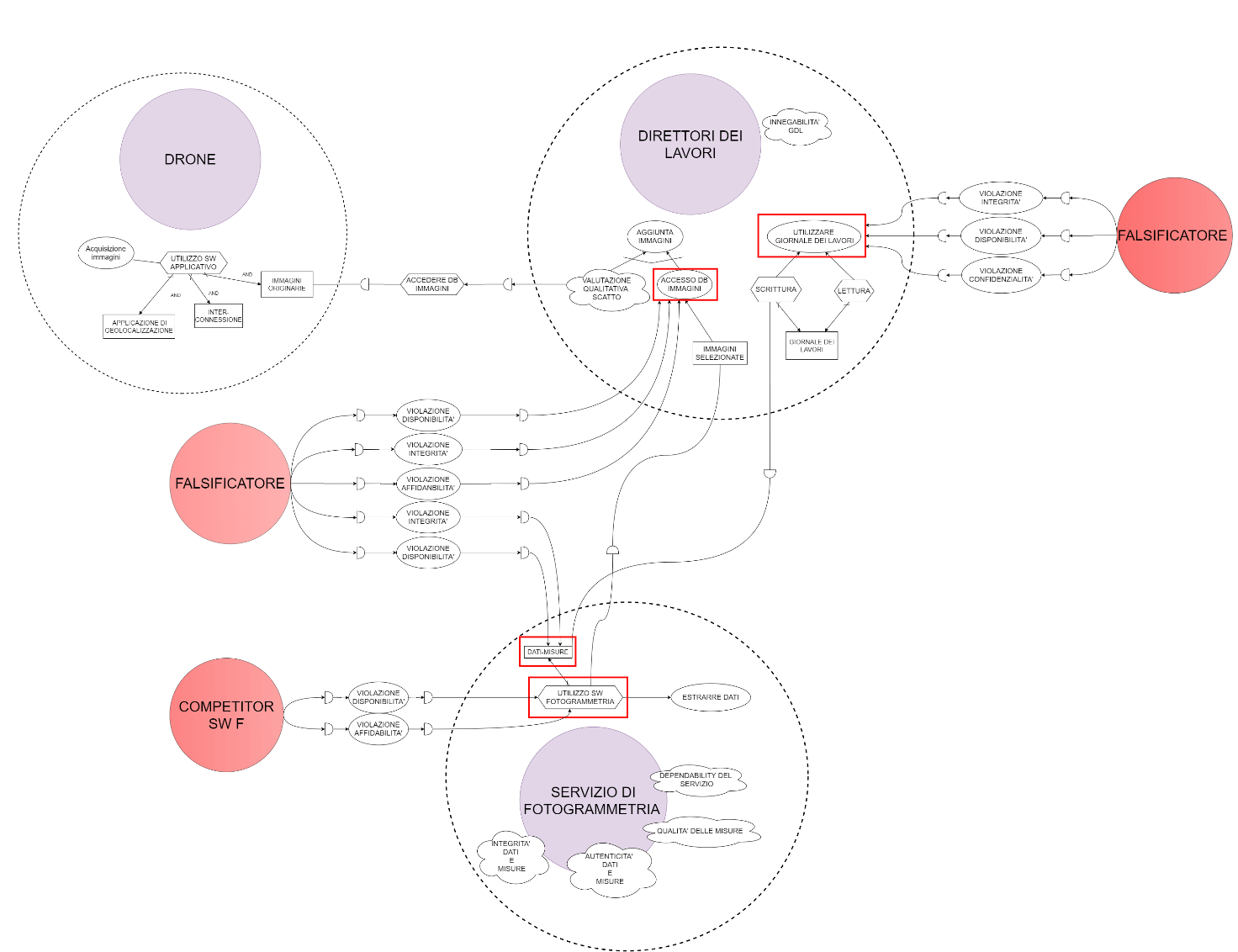
Andiamo ora a vederne l’inserimento nella rappresentazione del sistema effettuata tramite il modello i\*. Per rendere la visualizzazione più chiara ed esplicita abbiamo diviso i casi in cui si trattano attacchi malevoli *(figura 3)* ed i casi in cui si trattano attacchi dovuti a distrazioni *(figura 4)*.

Figura 3

*Grafico*

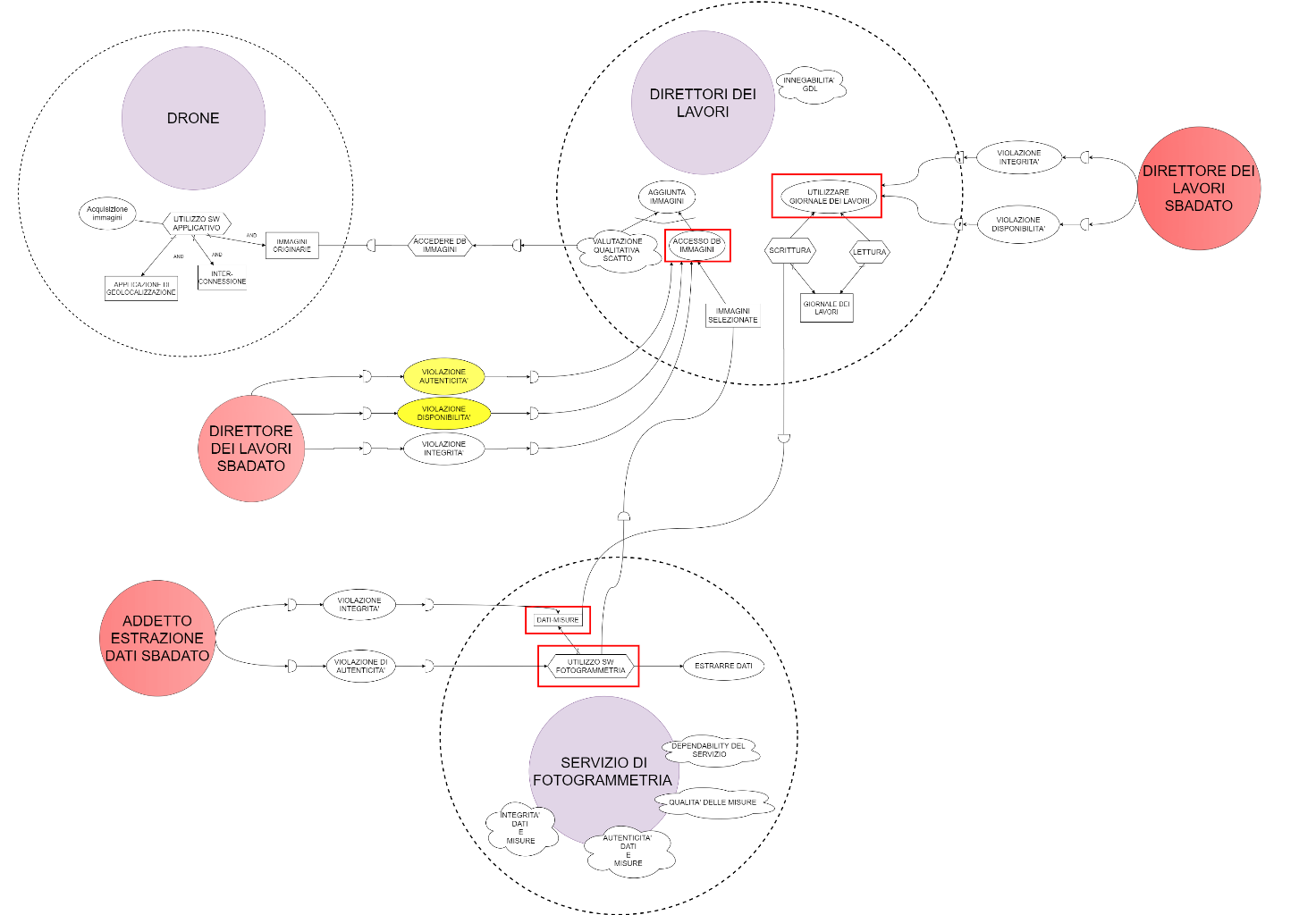


Figura 4

Nei nostri grafici abbiamo identificato i seguenti attaccanti:

* *Falsificatore*: si tratta di colui che cerca di alterare i dati interni al database delle immagini ed interni al giornale dei lavori, facendolo con lo scopo di danneggiare l’azienda, oppure per coprire errori commessi dai lavoratori;
* *Competitor del servizio di fotogrammetria*: è un servizio di fotogrammetria esterno, non selezionato per collaborare con l’impresa edilizia, che cerca di rendere improprio il servizio adottato per trovare le misure;
* *Direttore dei lavori sbadato*: si tratta del direttore dei lavori che, per distrazione o per incompetenza, compie degli errori che potrebbero rendere vulnerabile le superfici di attacco spianando così la strada ad un attaccante esterno.
* *Addetto all’estrazione dei dati sbadato*: si tratta dell’addetto all’estrazione dei dati, presenti all’interno del database delle immagini che poi verranno inseriti come parametri per ottenere le misure tramite il servizio di fotogrammetria, che commette degli errori involontari ma rilevanti che potrebbero facilitare gli attacchi.

Andiamo ora ad analizzare tutte le singole violazioni possibili, valutandone però in seguito le azioni di mitigazione.

|  |  |
| --- | --- |
| Use Case ID: AT-01 Use case Name: Violazione disponibilità | |
| Attori | Direttore dei lavori sbadato, Falsificatore |
| Descrizione | Una violazione di disponibilità comporta, accidentalmente o in modo illecito, la perdita dell’accesso o una cancellazione dei dati presenti nel database delle immagini. |
| Asset | Database Immagini |
| Stimolo e Pre-condizioni | Danneggiare l’azienda con l’accesso al database delle immagini o necessità di avere un ritardo sull’estrazione dei dati |
| Tipo di attacco 1 | Il direttore dei lavori elimina alcune immagini contenute nel database delle immagini selezionate tra tutte quelle ottenute dal drone per essere indirizzate al software di fotogrammetria |
| Tipo di attacco 2 | Il falsificatore compromette temporaneamente l’accesso e il recupero delle immagini presenti all’interno del database |
| Risposta e Post-condizioni | Database delle immagini compromesso o non accessibile |
| Mitigazioni | *valutati di seguito* |
| Requisiti non funzionali | *valutati di seguito* |

|  |  |
| --- | --- |
| Use Case ID: AT-02 Use case Name: Violazione integrità | |
| Attori | Direttore dei lavori sbadato, falsificatore |
| Descrizione | Una violazione di integrità comporta una compromissione dei valori, rendendo questi ultimi non più veritieri. |
| Asset | Database Immagini |
| Stimolo e Pre-condizioni | Accesso al database |
| Tipo di attacco 1 | Il direttore dei lavori elimina alcune immagini contenute nel database delle immagini selezionate tra tutte quelle ottenute dal drone per essere indirizzate al software di fotogrammetria o commette errori nel procedimento di salvataggio dei dati compromettendo il file |
| Tipo di attacco 2 | Il falsificatore compie azioni che rendono inaffidabili le immagini. Per esempio, modificando degli attributi delle immagini, quali ad esempio la data |
| Risposta e Post-condizioni | Database delle immagini compromesso |
| Mitigazioni | *valutati di seguito* |
| Requisiti non funzionali | *valutati di seguito* |

|  |  |
| --- | --- |
| Use Case ID: AT-03 Use case Name: Violazione autenticità | |
| Attori | Direttore dei lavori sbadato |
| Descrizione | Una violazione di autenticità comporta la modifica di dati rendendoli incongruenti con la realtà. |
| Asset | Database Immagini |
| Stimolo e Pre-condizioni | Accesso al database |
| Tipo di attacco 1 | Il direttore dei lavori approva delle immagini non qualitativamente buone all’interno del database |
| Tipo di attacco 2 | Il direttore dei lavori termina il proprio lavoro senza salvare le modifiche |
| Risposta e Post-condizioni | Immagini nel database compromesse |
| Mitigazioni | *valutati di seguito* |
| Requisiti non funzionali | *valutati di seguito* |

|  |  |
| --- | --- |
| Use Case ID: AT-04 Use case Name: Violazione Integrità | |
| Attori | Addetto estrazione dei dati sbadato, falsificatore |
| Descrizione | Una violazione di integrità comporta una compromissione dei valori, rendendo questi ultimi non più veritieri. |
| Asset | Dati |
| Stimolo e Pre-condizioni | Modifica dei dati con annesso danno all’azienda |
| Tipo di attacco 1 | L’addetto all’estrazione dei dati rende inaffidabili i dati estratti dalle immagini salvando male le corrispondenze con le stesse |
| Tipo di attacco 2 | Il falsificatore modifica i dati minando il loro futuro utilizzo |
| Risposta e Post-condizioni | Giornale dei lavori compromesso |
| Mitigazioni | *valutati di seguito* |
| Requisiti non funzionali | *valutati di seguito* |

|  |  |
| --- | --- |
| Use Case ID: AT-05 Use case Name: Violazione Disponibilità | |
| Attori | Direttore dei lavori sbadato, falsificatore |
| Descrizione | Una violazione di disponibilità comporta, accidentalmente o in modo illecito, la perdita dell’accesso o una cancellazione delle informazioni presenti nel giornale dei lavori. |
| Asset | Giornale dei lavori |
| Stimolo e Pre-condizioni | Entrare in possesso delle credenziali per accedere al giornale dei lavori |
| Tipo di attacco 1 | Il direttore dei lavori elimina per sbaglio il proprio account e non riesce più ad accedere al giornale dei lavori |
| Tipo di attacco 2 | Il falsificatore entra in possesso delle credenziali necessarie per accedere al giornale dei lavori e le modifica escludendo il direttore dei lavori dall’accesso |
| Risposta e Post-condizioni | Impossibilità di accedere alle informazioni presenti nel giornale dei lavori |
| Mitigazioni | *valutati di seguito* |
| Requisiti non funzionali | *valutati di seguito* |

|  |  |
| --- | --- |
| Use Case ID: AT-06 Use case Name: Violazione integrità | |
| Attori | Falsificatore, direttore dei lavori sbadato |
| Descrizione | Una violazione di integrità comporta una compromissione dei valori, rendendo questi ultimi non più veritieri. |
| Asset | Giornale dei lavori |
| Stimoli e Pre-condizioni | Accesso al giornale dei lavori per voler screditare il direttore dei lavori o per voler alterare l’operato |
| Tipo di attacco 1 | Il falsificatore modifica i dati del giornale dei lavori per far si che si abbiano informazioni incoerenti che comportino il licenziamento del direttore dei lavori |
| Tipo di attacco 2 | Il direttore dei lavori, che lavora con più progetti in contemporanea, memorizza dati di altri progetti all’interno del progetto corrente |
| Tipo di attacco 3 | Il falsificatore va a modificare a suo piacimento i dati alterandoli rendendoli incorretti |
| Risposte e Post-condizioni | I dati del giornale dei lavori non rappresentano più la realtà |
| Mitigazioni | *valutati di seguito* |
| Requisiti non funzionali | *valutati di seguito* |

|  |  |
| --- | --- |
| Use Case ID: AT-07 Use case Name: Violazione autenticità | |
| Attori | Addetto estrazione dei dati sbadato |
| Descrizione | Una violazione di autenticità comporta la modifica di dati rendendolo incongruenti con la realtà. |
| Asset | Software di Fotogrammetria |
| Stimolo e Pre-condizioni | Accesso al servizio di fotogrammetria |
| Tipo di attacco 1 | L’addetto all’estrazione dei dati entra nel software di fotogrammetria e impone parametri non corretti per l’ottenimento dei dati necessari |
| Risposta e Post-condizioni | Dati non congruenti rispetto le immagini da cui sono estratti |
| Mitigazioni | *valutati di seguito* |
| Requisiti non funzionali | *valutati di seguito* |

|  |  |
| --- | --- |
| Use Case ID: AT-08 Use case Name: Violazione Disponibilità | |
| Attori | Competitor Software Fotogrammetria |
| Descrizione | Una violazione di disponibilità comporta, accidentalmente o in modo illecito, la perdita dell’accesso o una cancellazione dei dati presenti nel database delle immagini. |
| Asset | Software di Fotogrammetria |
| Stimoli e Pre-condizioni | Spingere i clienti del servizio di fotogrammetria a cambiare servizio |
| Tipo di Attacco 1 | Il competitor del servizio di fotogrammetria rende impossibile l’accesso al software di fotogrammetria scelto |
| Risposte e Post-condizioni | I clienti che utilizzano il software di fotogrammetria non potranno accedere temporaneamente al servizio |
| Mitigazioni | *valutati di seguito* |
| Requisiti non funzionali | *valutati di seguito* |

|  |  |
| --- | --- |
| Use Case ID: AT-09 Use case Name: Violazione Affidabilità | |
| Attori | Competitor Software Fotogrammetria |
| Descrizione | Una violazione dell’affidabilità comporta l’inattendibilità dei dati dovuta ad una modifica della funzionalità del sistema |
| Asset | Software di Fotogrammetria |
| Stimoli e Pre-condizioni | Accesso al file sorgente del software di fotogrammetria |
| Tipo Attacco 1 | Il competitor del software di fotogrammetria modifica le funzionalità del software di fotogrammetria scelto rendendolo inaffidabile ed in grado di produrre valori inconsistenti |
| Risposte e Post-condizioni | I dati estratti dalle immagini saranno inconsistenti e pertanto scartati |
| Mitigazioni | *valutati di seguito* |
| Requisiti non funzionali | *valutati di seguito* |

|  |  |
| --- | --- |
| Use Case ID: AT-10 Use case Name: Violazione Confidenzialità | |
| Attori | Falsificatore |
| Descrizioni | Una violazione di confidenzialità comporta una divulgazione dei dati e delle informazioni interne al progetto verso terze parti non autorizzate ad averli. |
| Asset | Giornale dei Lavori |
| Stimoli e Pre-condizioni | Accesso al giornale dei lavori |
| Tipo Attacco 1 | Il falsificatore condivide informazioni private contenute nel giornale dei lavori con terzi |
| Risposte e Post-condizioni | Informazioni private saranno condivise con utenti esterni non autorizzati ad avere tali informazioni |
| Mitigazioni | *valutati di seguito* |
| Requisiti non funzionali | *valutati di seguito* |

|  |  |
| --- | --- |
| Use Case ID: AT-11 Use case Name: Violazione Disponibilità | |
| Attori | Falsificatore |
| Descrizioni | Una violazione di disponibilità comporta, accidentalmente o in modo illecito, la perdita dell’accesso o una cancellazione dei dati presenti nel database delle immagini. |
| Asset | Dati |
| Stimoli e Pre-condizioni | Danni all’azienda |
| Tipo Attacco 1 | Il falsificatore elimina dal database tutte le credenziali per accedere ai dati, in modo da poterci accedere solo lui e modificare i dati come vuole |
| Risposte e Post-condizioni | L’accesso ai dati è negato per un tempo indefinito, pena dover estrarre nuovamente i dati dalle immagini |
| Mitigazioni | *valutati di seguito* |
| Requisiti non funzionali | *valutati di seguito* |

|  |  |
| --- | --- |
| Use Case ID: AT-12 Use case Name: Violazione Affidabilità | |
| Attori | Falsificatore |
| Descrizione | Una violazione dell’affidabilità comporta l’inattendibilità dei dati dovuta ad una modifica della funzionalità del sistema |
| Asset | Database delle immagini |
| Stimoli e Pre-condizioni | Accesso al database delle immagini danneggiando l’azienda |
| Tipo Attacco 1 | Il falsificatore modifica i dati presenti all’interno del database delle immagini rendendoli inaffidabili e non utilizzabili |
| Risposte e Post-condizioni | I dati presenti all’interno del database delle immagini non sono più veritieri |
| Mitigazioni | *valutati di seguito* |
| Requisiti non funzionali | *valutati di seguito* |

### ATTACK TREE

La rappresentazione così dettagliata degli Abuse e Misuse Case ha permesso l’identificazione di Attack Tree dando la possibilità così di visualizzare tutte le superfici di attacco, ovvero i punti più vulnerabili del sistema che potrebbero essere utilizzati dall’attaccante come punto di “ingresso”.

Andiamo ora ad analizzare gli Attack Tree di ogni singolo malintenzionato in relazione all’asset di riferimento:

* *Addetto all’estrazione dei dati sbadato:*

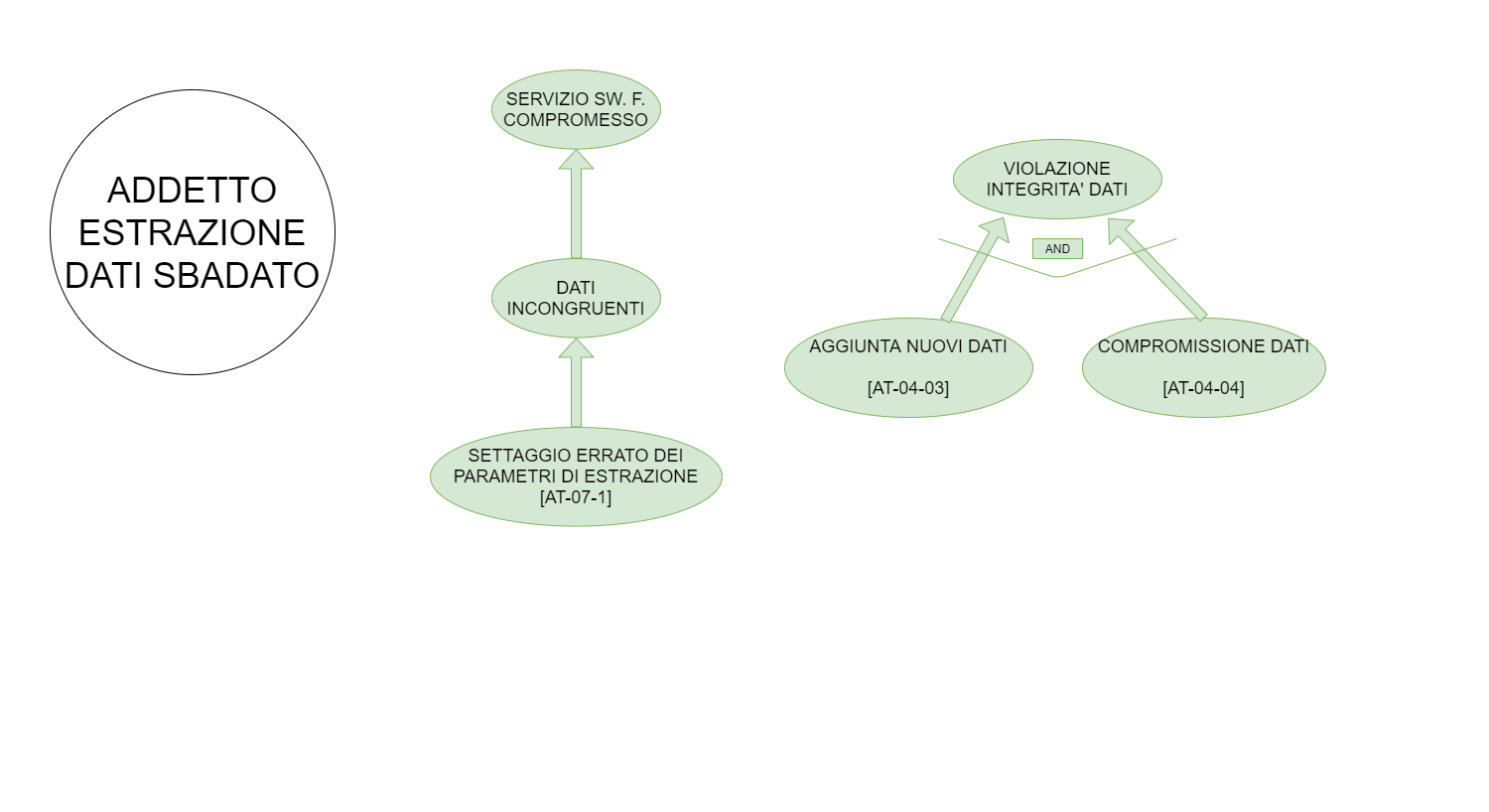
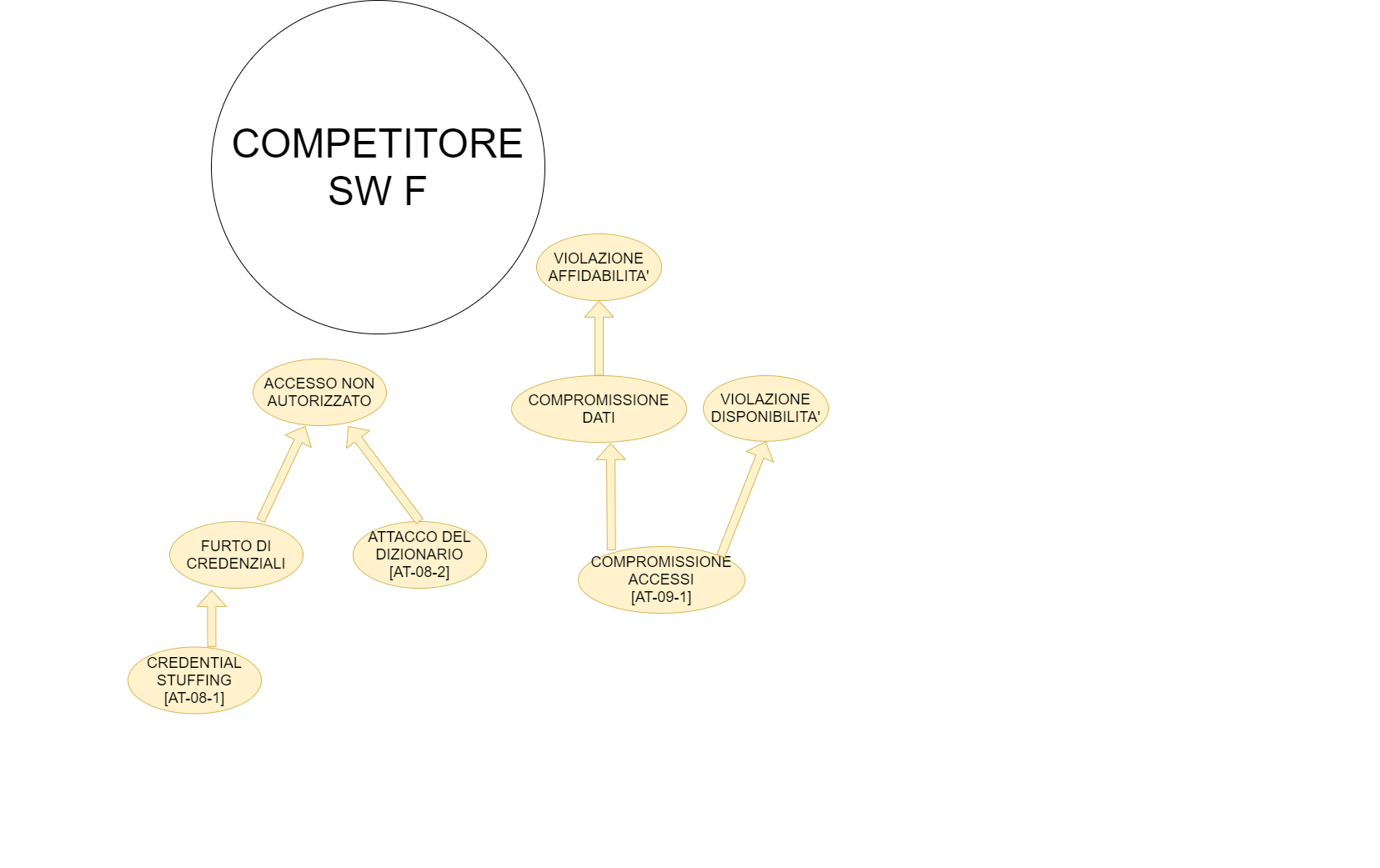
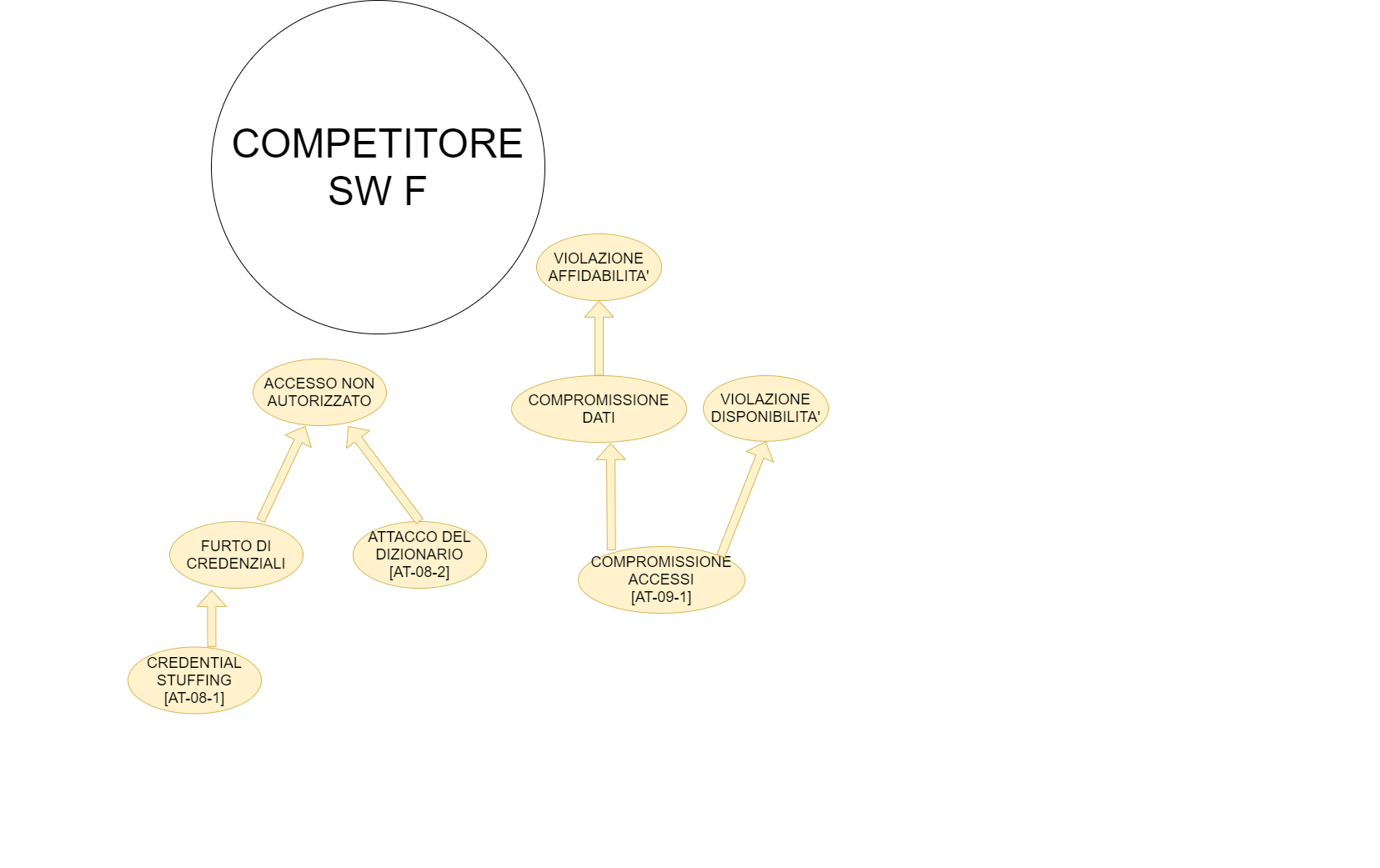
lavora esclusivamente con l’asset “dati”; i possibili pattern d’attacco possono essere inerenti al settaggio errato di parametri del servizio di fotogrammetria. In questo modo rende incongruente l’aggiunta o la modifica di alcuni dati presenti nel database compromettendo l’integrità.

Figura 5

* *Competitor del servizio di fotogrammetria:*

l’asset a cui fa riferimento tale attaccante è il servizio di fotogrammetria; i possibili pattern di attacco riguardano la violazione dell’affidabilità, possibile nel caso in cui vengono compromessi gli accessi ed i dati. La violazione della disponibilità dei dati può accadere nel caso in cui vengono violate le credenziali di accesso; la violazione della confidenzialità nel caso in cui vengono rubate delle password o nel caso di attaccato del dizionario, nel caso in cui le credenziali di accesso siano parole in una certa lingua, con qualche modifica.

Figura 6



* *Direttore dei lavori sbadato*

Facciamo ora riferimento a due asset: giornale dei lavori (GDL) e database delle immagini. Per quanto riguarda il *database delle immagini*, le violazioni che si possono verificare sono molte:

* *violazione dell’autenticità delle immagini*, come nel caso in cui vengono memorizzate immagini non nitide ed ottimali o nel caso in cui immagini buone vengano memorizzate in una posizione non corrispondente alla posizione geografica nel quale essa è stata rilevata;
* *violazione dell’integrità delle immagini*, causata ad esempio dal salvataggio di immagini errate o dalla compromissione delle immagini stesse;
* *violazione disponibilità delle immagini*, dovuta ad una chiusura improvvisa ed inaspettata del database o dall’eliminazione involontaria delle immagini.

Per quel che riguarda il giornale dei lavori si può avere una violazione disponibilità dei dati originari, dovuta essenzialmente a modifiche del giornale dei lavori non accuratamente valutate; in tale violazione è incorporata anche la violazione dell’integrità dei dati.

Figura 7

* *Falsificatore*

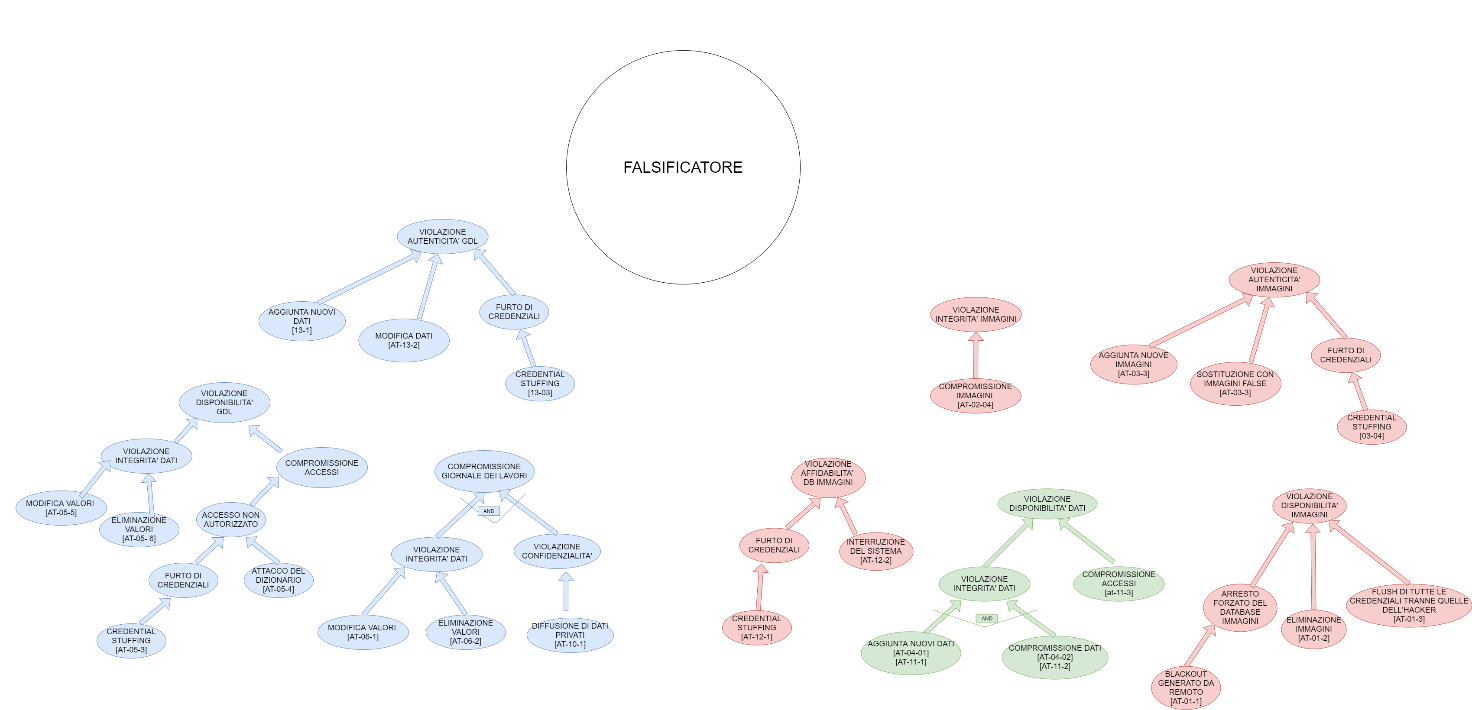
Le violazioni che può compiere il falsificatore ricadono sulla maggior parte degli asset del nostro sistema: giornale dei lavori, database delle immagini e dati.

Per quanto riguarda il giornale dei lavori le violazioni che si possono verificare sono:

* *violazione autenticità dei dati*: presentabile nel caso in cui vengono modificati, inseriti dati non congrui o nel caso in cui si verifichi un furto di credenziali;
* *violazione dell’integrità dei dati*, comportata da modifiche o di eliminazioni di importanti informazioni;
* *violazione delle disponibilità*: possibile a causa di un furto delle credenziali o di un attacco al dizionario che implica la compromissione degli accessi;
* *violazione di confidenzialità*: causata dalla diffusione volontaria di informazioni interne all’azienda a terze parti.

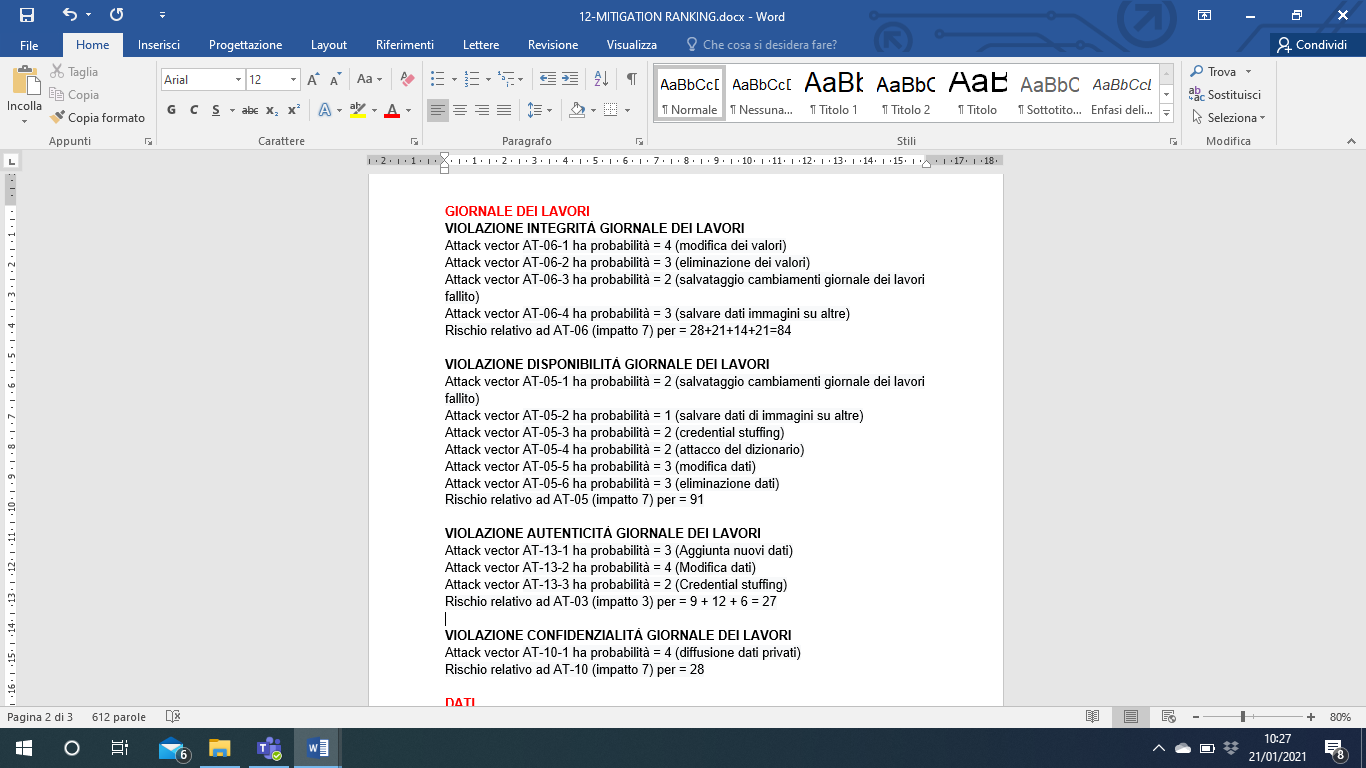
Trattando l’asset relativo ai dati prelevati dal servizio di fotogrammetria potremmo dire che le violazioni sono essenzialmente due: violazione dell’integrità dei dati, causata dall’aggiunta di nuovi valori e alla compromissione dei dati stessi, e la violazione della disponibilità dei dati, che comprende al suo interno l’integrità, dovuta alla compromissione degli accessi.

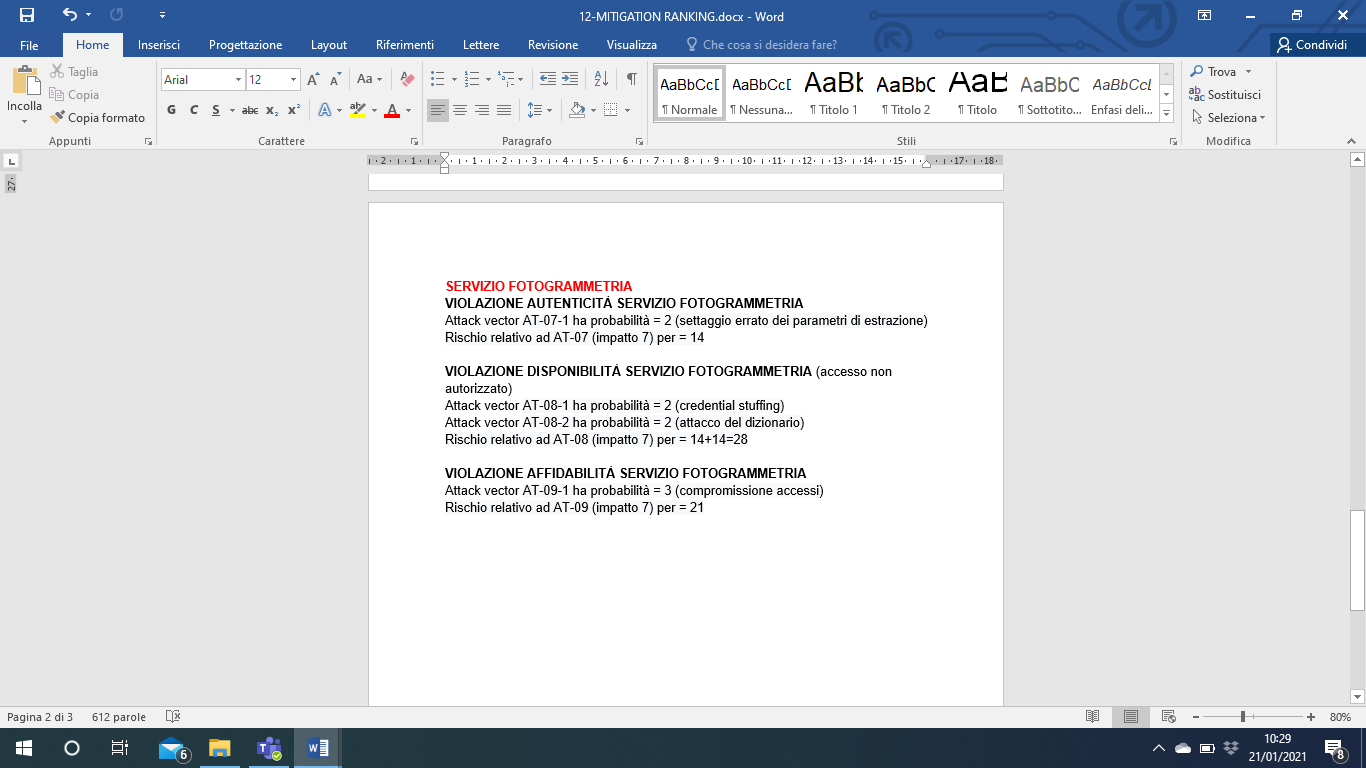
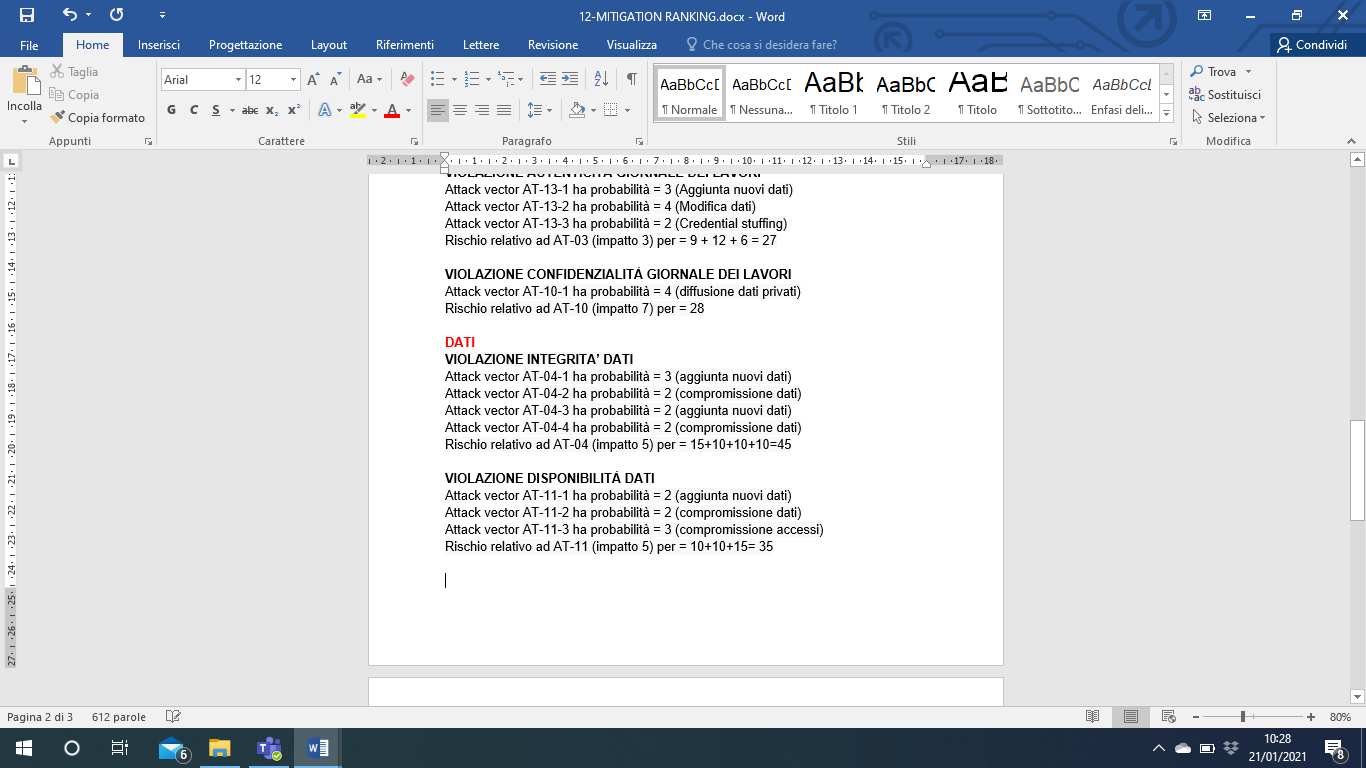
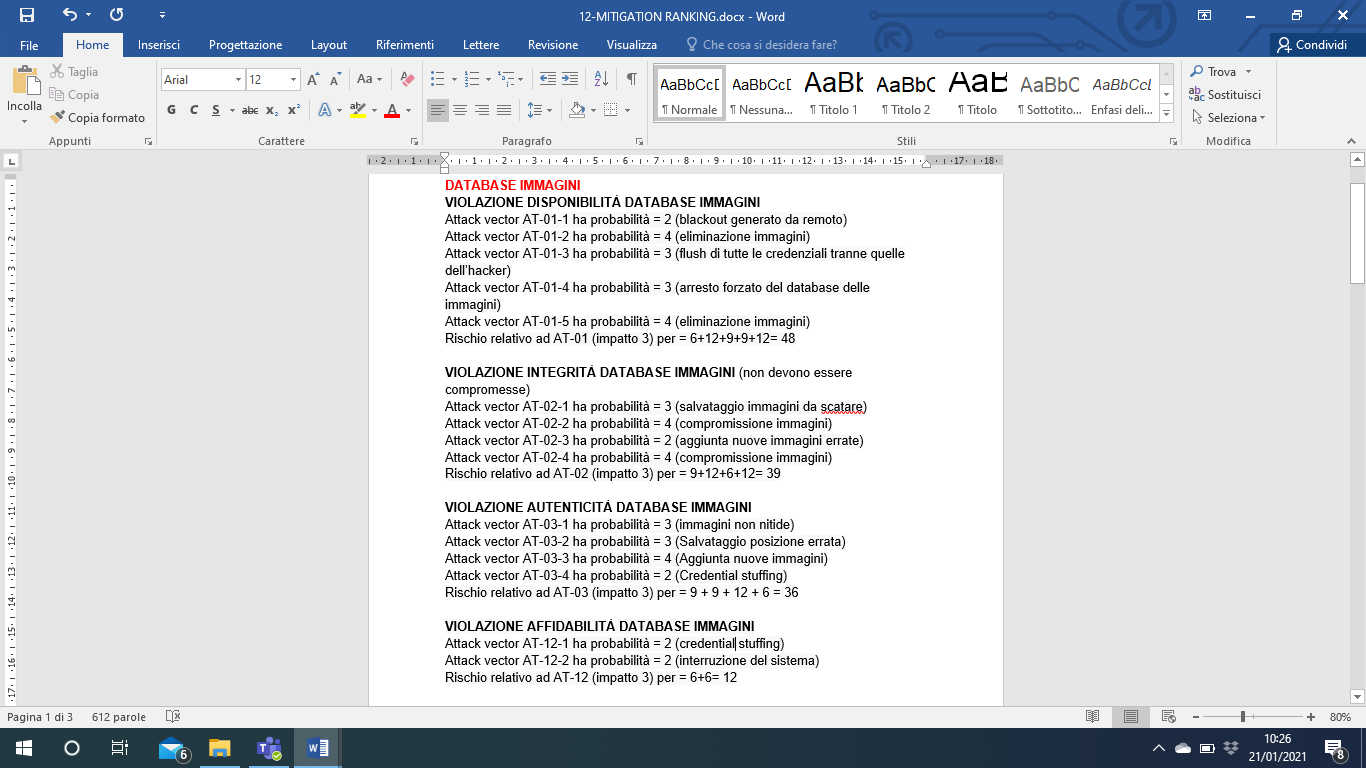
Un alto asset molto colpito dai possibili attacchi effettuati dal falsificatore è il database delle immagini. In questo caso le violazioni sono numerose:

* *violazione autenticità delle immagini*: nel caso in cui vengano aggiunte nuove immagini, nel caso in cui vengano sostituite immagini con immagini falsificate o quando vengono rubate le credenziali per poter accedere al database e manometterlo;
* *violazione dell’integrità delle immagini*: dovuta ad una compromissione dei dati, una modifica o ad una eliminazione;
* *violazione disponibilità delle immagini*, causata dall’eliminazione di immagini o dall’arresto forzato del database o dal blocco di tutte le credenziali ad eccezione di quelle utilizzate dall’attaccante;
* *violazione dell’affidabilità delle immagini*: provocata da un furto delle credenziali o da un’interruzione del sistema non prevista.

# ANALISI DEL RISCHIO

Fatta l’analisi dei rischi e la loro decomposizione il prossimo passo è quello di andare a mitigare il rischio. Si sfrutta il modello STRIDE (Spoofing, Tampering, Repudiation, Information Disclosure, Denial of Service, Elevation of Privilege) con in aggiunta DUA (Danger, Unreliability e Absence of Resilience).

Per ogni percorso di attacco, visualizzabile nell’attack tree, si deve calcolare la probabilità che si verifichi ogni attacco, e la si moltiplicarla all’impatto dell’asset relativo.



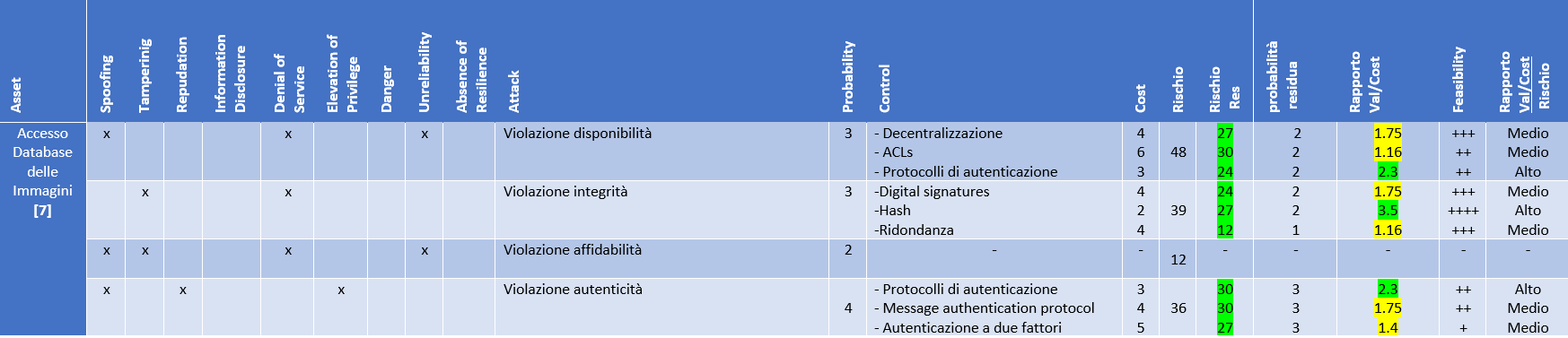
Facendo ciò andiamo ad ottenere un valore qualitativo che rappresenta il rischio relativo all’asset di riferimento, valore che ci permette di ragionare e lavorare in maniera concreta nelle azioni di mitigazione.

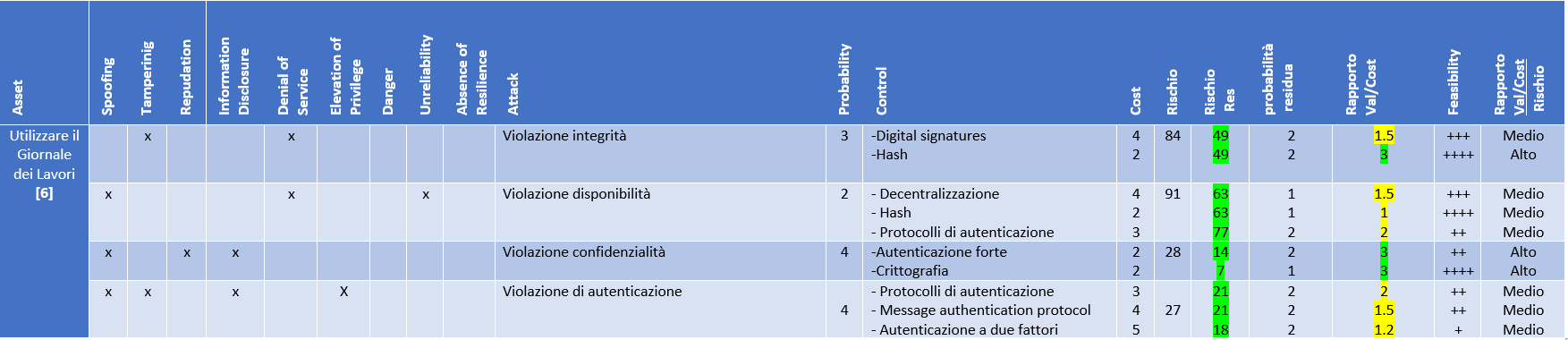
## ANALISI TECNICHE DI MITIGAZIONE

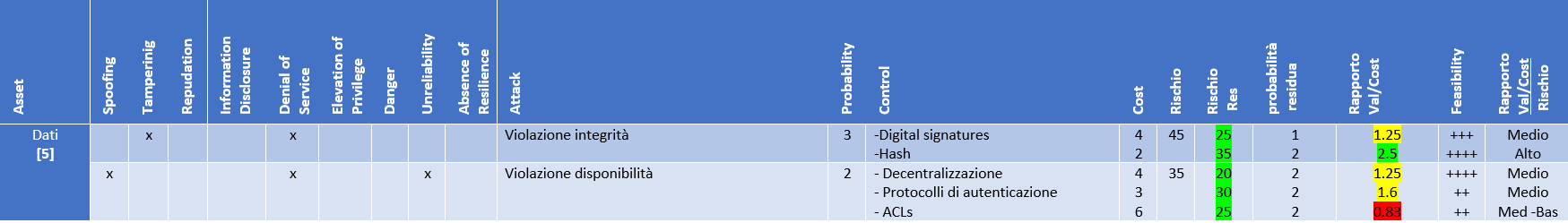
Per ogni possibile attacco, individuato nella fase precedente, vengono individuate una serie di strumenti e tecniche di mitigazione, in modo da ridurre il rischio e/o la probabilità che quella minaccia si concretizzi. Una volta fatto questo andremo a definire (in maniera qualitativa) il costo e la fattibilità che si ha nell’applicare quella specifica tecnica, analisi valore-costo ed infine, questo verrà rapportato con il rischio residuo.

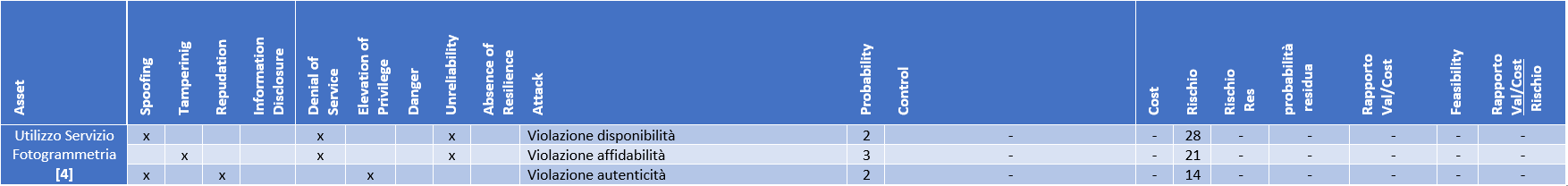
Questo ci permetterà di definire i requisiti di sicurezza.

Asset: Database delle immagini



Asset: Giornale dei lavori 

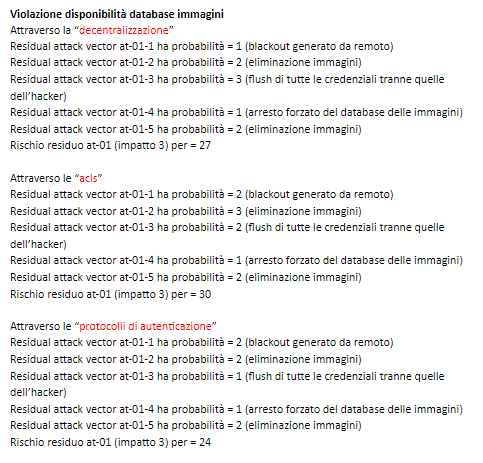
Asset: Dati 

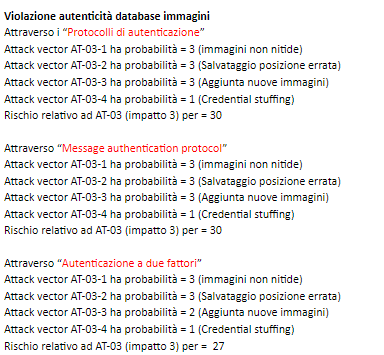
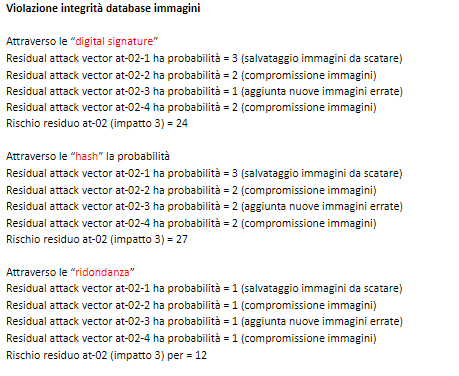
Asset: Software di fotogrammetria 

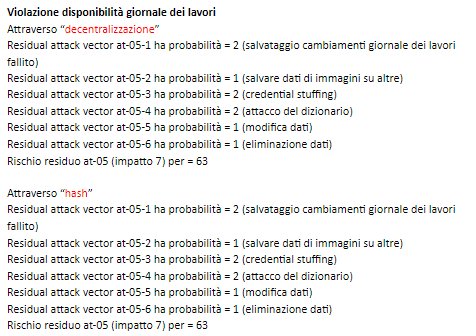
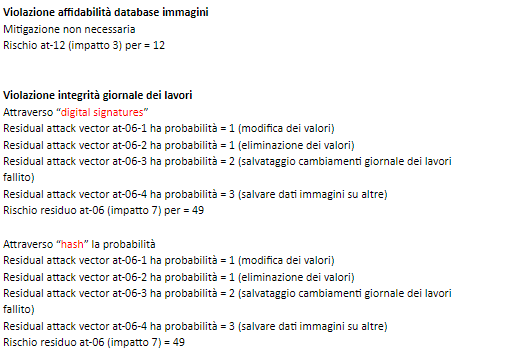
Nota 1: i valori delle probabilità residue (e non) sono stati calcolati come media di tutte le probabilità delle foglie del vettore di attacco.

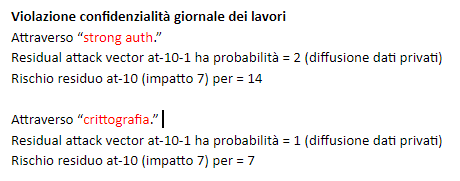
Nota 2: il valore della colonna “Feasibility” è tanto maggiore quanti “+” sono presenti.

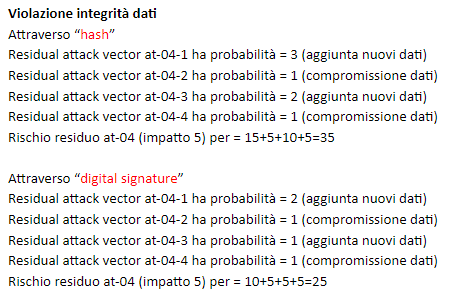
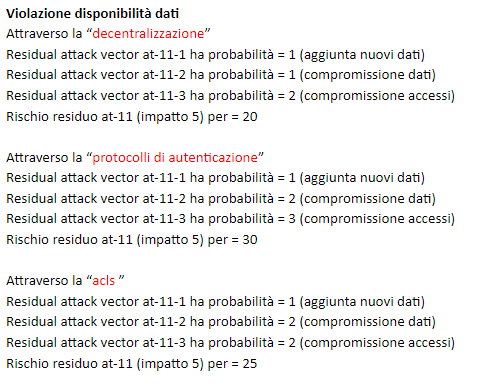
Nella pagina seguente sono riportati i calcoli che ci hanno permesso di completare le varie tabelle.

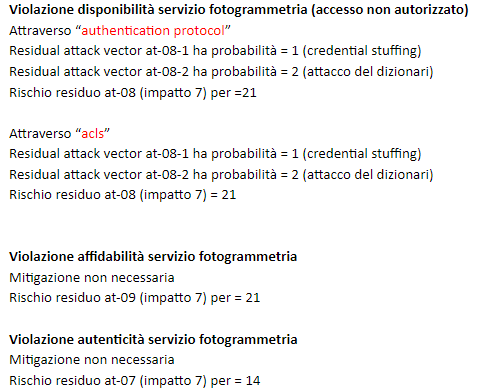
****

** **

****

****

** **

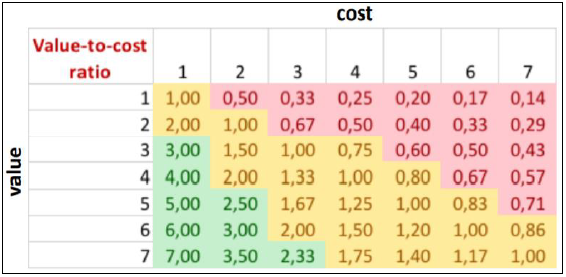
****

Al valore associato alla colonna “rischio residuo” è stato assegnato anche un colore che è relativo al tipo di rischio: partendo dall’attack vector si ha rischio massimo quando tutti i nodi delle foglie assumono probabilità pari a 7 (sempre considerando la scala qualitativa di likert a 7 valori); noi andiamo a confrontare il valore del rischio residuo e, in funzione del “range” di appartenenza, gli assegniamo un valore:

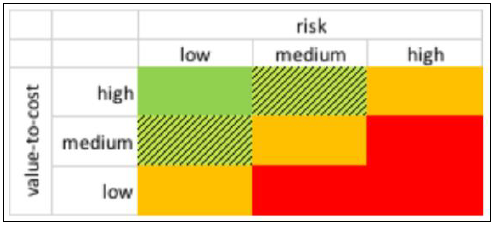
* verde se il rischio è inferiore al 33% del rischio massimo
* giallo se il rischio è compreso tra il 33% ed il 66% del rischio massimo
* rosso se il rischio è superiore al 66% del rischio massimo

Il valore della colonna “rapporto valore - costo” è stato calcolato nel seguente modo: per ogni tecnica di mitigazione si è calcolato il valore dell’asset a cui si riferisce rispetto il costo della singola tecnica; ovviamente maggiore è il valore e meglio è.

Anche a questo abbiamo aggiunto un colore: questo fa riferimento alla tabella sottostante e ci serve per assegnare una metrica a questa tecnica, in relazione appunto all’asset a cui viene applicata e al costo che si ha per implementarla.



Il valore della colonna “Rapporto Val/Cost – Rischio” è stato ottenuto dalla tabella sottostante: prendiamo i valori ottenuti prima (assegnando il colore), le corrispondenze riga – colonna ci danno un indice di quanto è profittevole quella tecnica per mitigare quella minaccia.



A valle delle analisi effettuate, per attenuare la probabilità degli attacchi sopra citati, e di conseguenza il rischio in funzione, si è scelto, per:

Database delle immagini:

* Violazione disponibilità: Decentralizzazione
* Violazione integrità: Hash
* Violazione affidabilità: mitigazione non necessaria
* Violazione autenticità: protocollo di autenticazione

Giornale dei lavori:

* Violazione integrità: Hash delle informazioni sul giornale dei lavori
* Violazione disponibilità: decentralizzazione
* Violazione affidabilità: protocollo di autenticazione
* Violazione confidenzialità: crittografia
* Violazione integrità (addetto dei lavori sbadato): doppia conferma
* Violazione disponibilità (addetto dei lavori sbadato): doppia conferma

Dati:

* Violazione integrità: Hash
* Violazione disponibilità: Mirroring
* Violazione integrità (addetto dei lavori sbadato): doppia conferma

Servizio di fotogrammetria:

* Non necessaria mitigazione, ci limitiamo ad utilizzare un servizio leader del mercato, con la quale ci interfacceremo e delegheremo i relativi problemi di disponibilità, affidabilità del servizio ed integrità delle misure.

Si prevede comunque una formazione e sensibilizzazione del personale per quanto riguarda l’utilizzo degli strumenti e degli account con cui si interfacceranno.

DESIGN

Blockchain

In questa fase si vanno a scegliere le tecnologie da utilizzare, in modo da soddisfare i requisiti di sicurezza definiti nella fase precedente. Si è ritenuto vantaggioso e conveniente utilizzare la Blockchain.

La blockchain è una sottofamiglia di tecnologie in cui il registro è strutturato come una catena di blocchi contenenti le transazioni e la cui validazione è affidata a un meccanismo di consenso, distribuito su tutti i nodi della rete, nel caso delle blockchain permissionless, o pubbliche o su tutti i nodi i nodi che sono autorizzati a partecipare al processo di validazione delle transazioni da includere nel registro nel caso delle blockchain permissioned o private.

Le principali caratteristiche delle tecnologie blockchain sono l’immutabilità del registro, la trasparenza, tracciabilità delle transazioni e la sicurezza basata su tecniche crittografiche.

La blockchain è basata su una rete e dal punto di vista delle funzionalità permette di gestire un database in modo distribuito. Dal punto di vista operativo è un'alternativa agli archivi centralizzati e permette di gestire l’aggiornamento dei dati con la collaborazione dei partecipanti alla rete e con la possibilità di avere dati condivisi, accessibili, distribuiti presso tutti i partecipanti. Di fatto permette una gestione dei dati in termini di verifica e di autorizzazione senza che sia necessaria una autorità centrale.

La blockchain è da vedere come una piattaforma che consente lo sviluppo e la concretizzazione di una nuova forma di rapporto sociale, che grazie alla partecipazione di tutti è in grado di garantire a tutti la possibilità di verificare, di “controllare”, di disporre di una totale trasparenza sugli atti e sulle decisioni, che vengono registrati in archivi che hanno caratteristica di essere inalterabili, immodificabili e dunque immuni da corruzione.

La blockchain può essere considerata una tecnologia che appartiene alla categoria delle tecnologie Distributed Ledger, archivi distribuiti. Le Distributed Ledger Technology o DLT possono essere definite come un insieme di sistemi caratterizzati dal fatto di fare riferimento a un registro distribuito, governato in modo da consentire l’accesso e la possibilità di effettuare modifiche da parte di più nodi di una rete.

Qualunque transazione, ovvero i dati che la rappresentano, è sottoposta ad un meccanismo di firma a doppia chiave asimmetrica che, pur non dotata di certificati rilasciati da certificatori accreditati (la blockchain prevede appunto il superamento di organismi certificatori centralizzati), funziona con un meccanismo simile a quello della firma digitale. Le DLT prevedono l’utilizzo di algoritmi crittografici che abilitano l’utente all’utilizzo del sistema mettendogli a disposizione una chiave pubblica ed una privata che viene usata per sottoscrivere le transazioni o per attivare gli smart contract o altri servizi collegati alla blockchain.

Le blockchain sono delle Distributed Ledger technology caratterizzate da un registro impostato e strutturato in modo da gestire le transazioni all’interno di una catena di blocchi. Dal punto di vista delle “regole di gestione”, ciascun blocco si “aggiunge” alla catena sulla base di un processo basato sul Consenso distribuito su tutti i nodi della rete, ovvero con la partecipazione di tutti i nodi che vengono chiamati a contribuire alla validazione delle transazioni presenti in ciascun blocco e alla loro “inclusione” nel registro.

La Blockchain è un grande database per la gestione di transazioni crittografate su una rete decentralizzata di tipo peer-to-peer che dà il nome ad una nuova piattaforma tecnologica, che permette di ridefinire e reimpostare il modo in cui creiamo, otteniamo e scambiamo valore. La Blockchain sta facendo con le transazioni quello che Internet ha fatto con le informazioni e lo sta facendo grazie ad un processo che unisce sistemi distribuiti, crittografia avanzata e teoria dei giochi.

Smart contract

Il significato letterale di smart contract è “contratti intelligenti”. In concreto, sono software basati sulla tecnologia blockchain.

Essi definiscono regole e penali di un accordo allo stesso modo di un contratto tradizionale. La differenza sostanziale consiste nella presenza nel software delle funzioni if/then, le quali rendono automatico il pagamento al verificarsi di una determinata condizione.

Più in generale, gli smart contract aiutano le persone a scambiare denaro, trasferire proprietà e qualsiasi altra cosa di valore in modo trasparente e senza ricorrere ai servizi di un intermediario.

Il primo passaggio è la stipula di un contratto. Le due parti trascrivono poi le clausole in uno smart contract.

Successivamente, il “contratto intelligente” viene inserito nella blockchain. I partecipanti (i nodi della catena di blocchi) verificano l’esattezza delle informazioni riportate, quindi l’accordo digitale entra a far parte di un blocco.

A questo punto, gli utenti della blockchain validano il blocco tramite il meccanismo Proof of Work (PoW). Chi si occupa della validazione del blocco riceve a sua volta la criptovaluta Ether come ricompensa.

Terminata l’operazione, il blocco contenente lo smart contract viene aggiunto alla blockchain. Da qui in avanti la responsabilità di monitorare che le condizioni del contratto si verifichino nella realtà, è affidata a un terzo agente (come, ad esempio, un’applicazione per smartphone).

Non appena l’app invia il segnale alla blockchain che una o più condizioni si sono compiute, quest’ultima rende automatica l’esecuzione dello smart contract.

Uno dei vantaggi degli smart contract è l’autonomia rispetto a un’entità terza (intermediario) e conseguente risparmio di denaro e tempo, sicurezza dei dati e un’accuratezza maggiore rispetto ai contratti tradizionali.

Un secondo vantaggio è una diretta conseguenza della prima. L’impiego di un intermediario comporta un onere a carico di una o entrambe le persone che stipulano un accordo.

La blockchain è uno dei più grandi contributi alla sicurezza dei dati. Per questo motivo, i “contratti intelligenti” e le informazioni contenute in essi sono molto più sicuri rispetto a quelli tradizionali.

Gli elementi che vanno a costituire le condizioni contrattuali sono crittografati, così come succede ad esempio per i messaggi scambiati ad esempio sull’applicazione [Telegram](https://www.money.it/Telegram-cos-e-come-funziona).

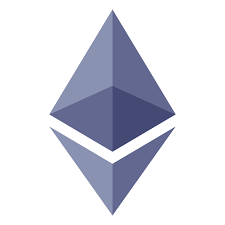
I contratti compilati a mano possono essere soggetti a errori o difetti che minano il contenuto stesso dell’accordo. Una situazione che, invece, non si verifica in ambiente informatico. Da questo punto di vista gli smart contract possiedono un’affidabilità sensibilmente maggiore.

# IMPLEMENTAZIONE

## TECNOLOGIE PER L’IMPLEMENTAZIONE

In questa fase andiamo a descrivere tutte le tecnologie utilizzate per realizzare il progetto finale: un’applicazione web che permette di andare a caricare le immagini provenienti dal drone, corredandole di ‘dati’ e ‘metadati’ tenendo conto dei requisiti di sicurezza analizzati precedentemente.

### ETHEREUM e QUORUM

Ethereum è una blockchain centralizzata open source la cui funzionalità principale è quella di generare dei contratti intelligenti. Essa utilizza una criptovaluta nativa dalla piattaforma stessa nota come Ether (ETH) ed è la seconda più grande e più utilizzata dopo i Bitcoin.

Da essa è nata la piattaforma in grado di generare contratti intelligenti privati, ovvero regolati da una rete di autorizzazioni, nota come Quorum. Noi abbiamo utilizzato questa “fork” di Ethereum perché permette di implementare blockchain private e permissioned in grado di garantire:

* Privacy
* Permessi di rete
* Performance migliori

### SOLIDITY e JAVASCRIPT

Solidity è un linguaggio di programmazione orientato agli oggetti e di alto livello utilizzato per scrivere smart contract su varie piattaforme per le blockchain, in particolare con Ethereum e con la sua Virtual Machine; inoltre, abbiamo utilizzato il linguaggio JavaScript che ci ha permesso di comunicare con la blockchain e di implementare l’interfaccia web del lato client.

### NODE.JS e EXPRESS

Node.js è un runtime environment open source e multipiattaforma che consente di implementare applicazioni web lato server grazie all’utilizzo di JavaScript. Tale ambiente presenta numerosi vantaggi:

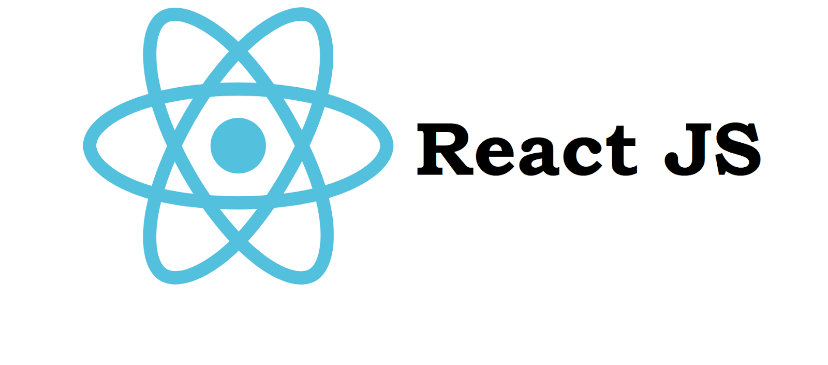
* Buone performance
* Scalabilità per le applicazioni web
* Introduce il concetto di Node Package Manager

Inoltre, per lo sviluppo della Web app, abbiamo deciso di affiancare a Node.js il framework Express, molto veloce ed in grado di fornire meccanismi per la gestione di pagine HTTP in base a diversi url noti come ‘route’. Express ci ha permesso di realizzare diverse pagine dell’applicazione e di implementare funzioni presenti in numerose librerie che consentono di lavorare con sessioni, accessi utente e molto altro.

### GANACHE

Ganache fa parte della Truffle Suite, ovvero un set di strumenti per sviluppatori che consente agli utenti di ricreare ambienti con blockchain personali e permette di testare smart contract. Si tratta dunque di una blockchain utilizzata per lo sviluppo e per il test di ciò che si genera.

### REACT

React è una libreria Javascript che permette di implementare interfacce utente complesse tramite la loro suddivisione in componenti più semplici. L’idea di base è dunque quella di permettere lo sviluppo di progetti grandi agevolando la progettazione, lo sviluppo ed il testing.

React, inoltre, supporta la definizione di modelli HTML per ogni componente del progetto iniziale tramite un approccio dichiarativo che viene aggiornato costantemente dal ‘core’ della libreria. Tutto ciò può avvenire sul server, tramite Node.js, oppure all’interno del browser sul client, permettendo ad interfacce molto complesse di aggiornarsi in tempo reale.

### MONGODB

MongoDB è un database NoSQL orientato ai documenti, utilizzato per l’archiviazione di grande dimensione di dati, che gode di molte funzionalità, tra cui quelle disponibili nei database tradizionali. Inoltre, invece di utilizzare tabelle con colonne e righe come nei database relazionali tradizionali, MongoDB fa uso di raccolte e documenti. Tali documenti sono costituiti da coppie-valore, il concetto base di MongoDB. I database NoSQL sono molto popolari grazie alla loro semplicità di controllo dei dati e al loro design sobrio.



### IPFS (InterPlanetary File System)

L’IPFS ha permesso di gestire le immagini e gli hash associati a queste. Si tratta di un protocollo di comunicazione ad una rete peer-to-peer per l’applicazione e la condivisione di dati in un file system distribuito. IPFS è costruito tramite un sistema decentralizzato composto da operatori che hanno una parte dei dati da condividere. Qualsiasi utente può salvare file grazie ad una tabella di hash distribuita messa a disposizione da IPFS.

## DETTAGLI IMPLEMENTATIVI

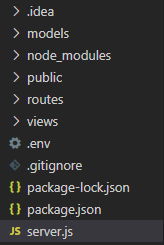
Possiamo dividere il codice da noi implementato in due parti principali: una riguardante lo sviluppo di un portale nel quale vengono immagazzinate le immagini e gli utenti ed una incentrata sulla realizzazione dello smart contract che permette di interfacciarsi con la blockchain.

Abbiamo pensato di implementare un database dal quale il direttore dei lavori interagisce, e può creare nuovi utenti che ricoprono il ruolo di dipendenti, attraverso la parte front-end.

Lo scopo di questo database è andare a salvare le immagini che provengono dal drone; tra queste poi verranno selezionate solo quelle approvate dal direttore dei lavori e attraverso lo smart contract salva sulla blockchain le sue informazioni. Possiamo poi andare a visualizzare l’immagine associata ad una determinata informazione attraverso l’uso di IPFS, fornendogli l’hash dell’immagine stessa.

Andiamo ora a spiegarne i punti salienti.

### SVILUPPO APPLICAZIONE WEB

I file implementati per ottenere l’applicazione sono molti ma i più rilevanti sono:

* *models*: cartella con le classi per la gestione del back-end dell’app relativo alla connessione con MongoDB, che permette di memorizzare le immagini e gli utenti
* *routes*: cartella con i file JavaScript che consentono di linkare più url, anche grazie all’utilizzo della libreria ‘express’
* *view*: cartella con tutti gli elementi delle ‘viste’, ovvero il posizionamento dei componenti nella schermata di visualizzazione.

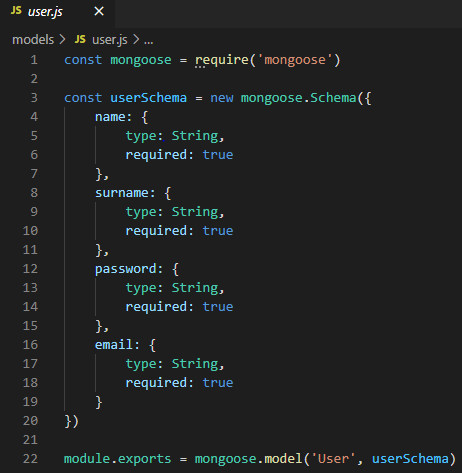
Analizziamo più nel dettaglio le tre cartelle principali:

* *MODELS*

Di seguito abbiamo riportato i file contenuti nella cartella *models* che permettono di memorizzare i dati nel database di MongoDB, sia se si tratta di un utente sia se si tratta di un’immagine.

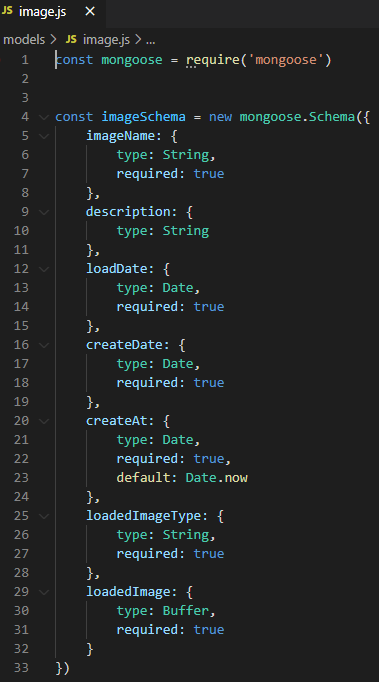
I codici presenti nella cartella sono essenzialmente due:

*user.js*

Questo script rappresenta la struttura degli utenti all’interno del database.

Inizialmente è stata importata la classe di MongoDB nota come ‘mongoose’. È stato poi inizializzato l’oggetto ‘userSchema’ contenente i nomi dei campi in grado di identificare un utente e il tipo di dato relativo a quel campo.

*image.js*



Il codice riportato di fianco si occupa della gestione delle immagini per poterle poi memorizzarle all’interno del database.

Anche in questo caso è stata importata la classe ‘mongoose’ e successivamente è stato generato l’oggetto relativo a tutte le immagini, con i suoi specifici campi e con l’identificazione della tipologia di dato, che verrà poi memorizzato nel database.

* *ROUTES*

Tale cartella contiene tre file JavaScript contente i codici che permettono di collegare le diverse pagine web tra loro, a seguito di azioni effettuate dall’utente; le principali sono:

*image.js*

In questo caso viene importato il file *image.js*, esplicitato precedentemente nella cartella *models*, e viene importato l’elemento ‘router’ interno alla libreria Express.

In questo script si implementa un codice che permette di gestire le azioni preformabili nell’applicazione, quali ad esempio l’esecuzione di query nel database, il caricamento di file, etc.

Un esempio potrebbe essere il passaggio dalla pagina di *home* alla pagina relativa alle visualizzazioni delle immagini presenti nel database o alla pagina per il caricamento di nuove immagini.

Inoltre, sono state sviluppate delle tecniche per la gestione di eccezioni, come un errato caricamento dei dati, grazie all’istruzione di ‘try … catch’. Queste anomalie saranno poi facilmente valutate e risolte dagli utenti grazie all’applicazione che invierà in output un messaggio di errore.

*index.js*

La logica di tale pagina è identica alla logica della pagina precedente, ma essa tratta il passaggio la pagina della *home* ed una pagina che visualizza le ultime immagini caricate nel database.

*users.js*

Anche in questo caso la natura del codice è la stessa delle altre, ma essa si occupa degli url relativi alle operazioni riguardanti gli utenti, come il caricamento di nuovi utenti o la visualizzazione della lista degli utenti registrati, etc.

* *VIEW*

Nella cartella v*iew* sono presenti tutti i codici html delle pagine web implementate, utilizzati anche dai codici contenuti nella cartella r*outes*. In questi file si stabilisce come apparirà graficamente la nostra applicazione: grandezza dei pulsanti, sfondo, box, button, …

Spieghiamo alcuni codici di questa cartella:

*layout*

contiene il file layout.ejs utile a settare lo stile delle finestre dell’applicazione (colore dello sfondo, posizione del titolo, posizione dei componenti della home, …).

*partials*

contiene i file html sviluppati per descrivere le componenti condivise tra tutte le pagine web visualizzabili dalla nostra applicazione, come i button, le textlabel, i link, …

*index.ejs*

serve per renderizzare a schermo le immagini aggiunte di recente su MongoDB

*Images*

contiene i file html per la gestione di tutte le operazioni relative alle immagini. Alcuni esempi sono la lettura delle immagini, il caricamento di immagini nuove, … Ciò è stato implementabile con delle ‘action’, di tipo POST o GET, all’interno delle “form”. Queste azioni vengono catturate dai file javascript nella cartella routes, che richiamano i metodi di “mongoose” per leggere o scrivere le immagini su database

*user*

contiene i file html per la gestione di tutte le operazioni relative agli utenti. Anche in questo caso alcuni degli esempi possono essere il caricamento di nuovi utenti, la visualizzazione della lista degli utenti già memorizzati, …

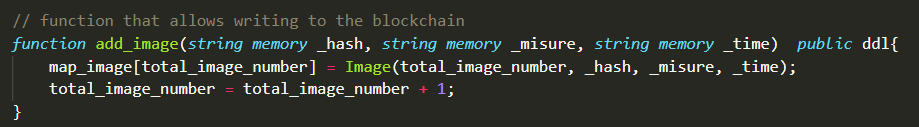
### SVILUPPO BLOCKCHAIN E SMART CONTRACT

Spieghiamo ora le parti più importanti del codice relativo all’utilizzo delle blockchain ed all’implementazione degli smart contract.

Nel nostro caso ci siamo serviti di un solo Smart contract “galma\_smart\_contract.sol”, utilizzato per caricare sulla blockchain le informazioni associate all’immagine, quali:

* ID univoco per l’immagine
* hash dell’immagine
* orario di caricamento sulla blockchain
* misure associate all’oggetto in esame nell’immagine

Abbiamo inoltre introdotto restrizioni in scrittura sulla blockchain attraverso la funzione “modifier”: permettiamo al solo direttore dei lavori l’inserimento delle immagini (ovvero le 4 info sopra citate) sulla blockchain. Questo è stato realizzando andando a rendere la funzione *“add\_image” public* per il solo direttore dei lavori, avendo assunto che sia lui a fare il deploy del contratto nella blockchain.



Sono state poi introdotte nello smart contract altre due funzioni:

- Una funzione che restituisce, dato l’ID dell’immagine, le informazioni associate a quell’id;

- Una funzione che restituisce il numero di immagini salvate nella blockchain.

Un’altra parte saliente del codice è quella contenuta nella cartella ‘src/components’ nel file *App.js.* Riportiamo quindi i blocchi di codice realizzati che sono di maggior rilevanza in questo file, commentandoli brevemente.



Qui andiamo a definire delle variabili che prenderanno valore durante l’esecuzione dell’app, in particolare:

Variabili fisse una volta che accedo con un determinato account alla blockchain

- Contract conterrà il contratto con cui ci si interfaccerà (galma\_contract.sol), utile per richiamare i suoi metodi;

- Account conterrà l’indirizzo dell’account con il quale l’user farà l’accesso su ganache

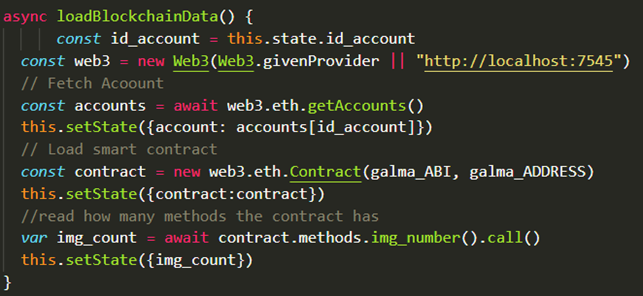
- Id\_account: variabile che viene settata durante la fase di login, e che identifica se un utente ha i privilegi in scrittura o in sola lettura

Variabili temporanee

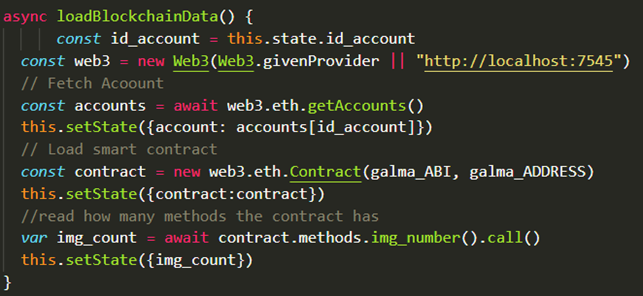
- Id, hash, misure, date: contengono le info dell’immagine caricata (quando carico l’immagine vengono mostrate in anteprima le sue info) o richiesta dal text-box.

- Isnum flag: per la sanificazione dell’input del text-box

- Img\_count: contatore del numero di immagini totali caricate

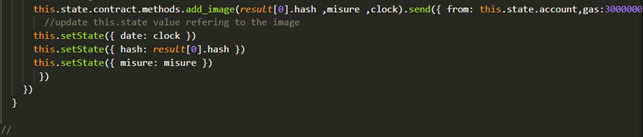


Con questo blocco di codice ci connettiamo a Ganache tramite il numero di account definito da “id\_account”, andiamo a settare la variabile contract ed estraiamo il numero di immagini caricate.



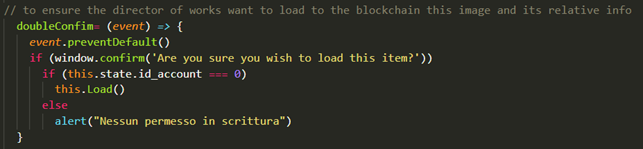
Questa è la funzione che permette di ricevere le informazioni associate all’immagine identificata da un certo id fornito in input(*value*).

Attraverso *contract.methods.get(value).call()* andiamo a richiamare il metodo *get* dello smart contract che ci restituisce il vettore con le 4 componenti richieste, che salviamo rispettivamente nelle variabili *id, hash, misure e date*.



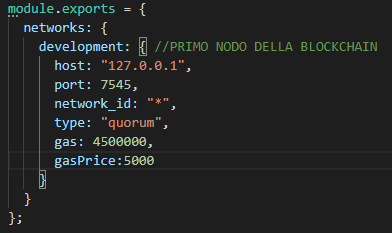
Questa funzione invece permette di caricare l’immagine (ovvero le sue informazioni) sulla blockchain. Per prima cosa si va a mettere l’hash della stessa in *result*.

Andiamo a calcolare le otto misure dell’immagine, che per semplicità abbiamo considerato valori randomici > 0 e il timestamp corrente. Quindi non resta che andare a richiamare il metodo “*add\_image”* e passargli i valori sopra descritti; questo permette di andare a cristallizzare le informazioni nella blockchain.



Per fare questo però abbiamo inserito una doppia conferma: non basta che il direttore dei lavori prema il tasto “invia” per caricare le informazioni, ma è stato inserito un controllo ulteriore per far sì che sia certo di doverla caricare e non abbia cliccato involontariamente il tasto.

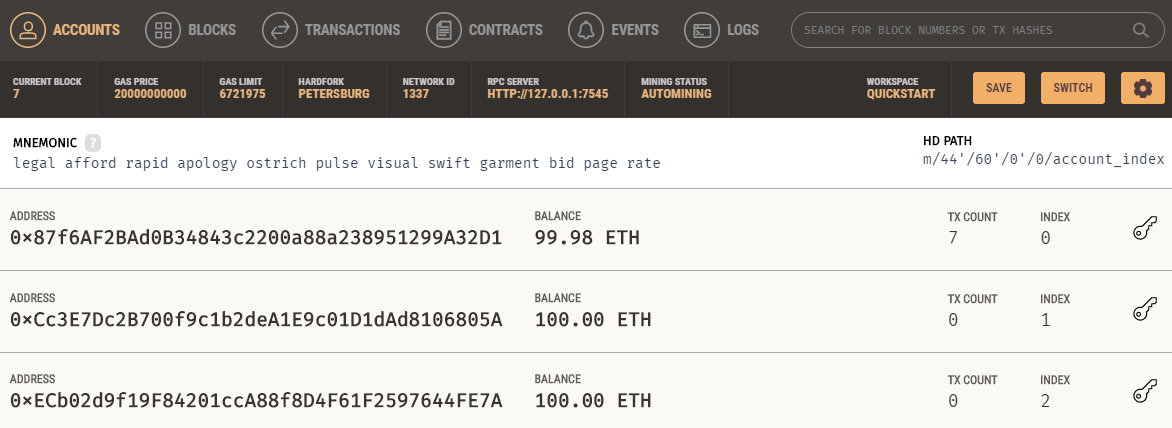
Per una maggior sicurezza si è deciso comunque di ridondare i controlli, questa volta per prevenire utenti malevoli, facendo in modo che il caricamento delle informazioni si fermi qualora l’account con cui si è connessi alla blockchain sia diverso da quello del direttore dei lavori (account 0).

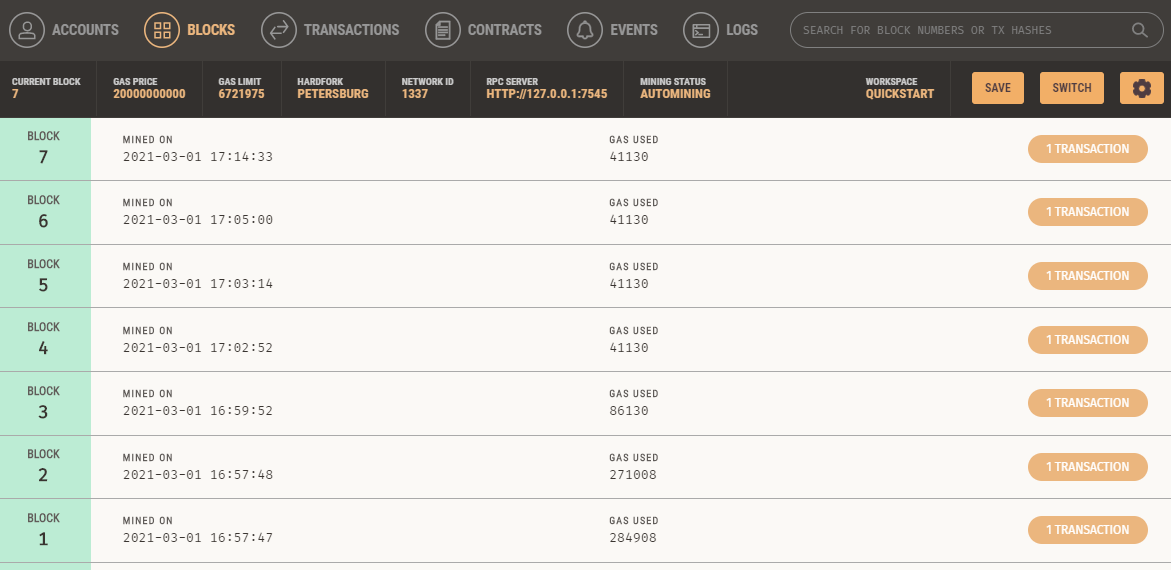


Infine, riportiamo il codice *Truffle-config.js*, ovvero il file chepresenta tutte le proprietà della blockchain, che vengono esplicitate da Ganache, e che permette l’iterazione tra la nostra Web-app e con Ganache. Questo risulta di rilevante importanza perché è Ganache che si occupa di ospitare la rete Ethereum e di salvare le transazioni sulla rete nei rispettivi blocchi. Si noti come, non essendo specificato alcun account,  
si considera che il deploy sulla blockchain dello smart contract sia effettuato dal primo indirizzo.

## IMPLEMENTAZIONE GANACHE

Avendo utilizzato l’applicazione di Ganache, riteniamo interessante esplicare quanto abbiamo fatto basandoci sulle schermate messe a disposizione dall’applicazione.

Qui si possono notare le reti Ethereum con i rispettivi indirizzi e con il valore di ETH che identifica la valuta associata al wallet dell’indirizzo.

In questa schermata sono raffigurati i blocchi che vengono registrati ogni volta che si effettua una transazione e nel quale vengono memorizzate le informazioni delle immagini approvate.



Se scendiamo più nel dettaglio ed espandiamo un singolo blocco possiamo trovare l’id della transazione a cui fa riferimento, la quantità di valuta usata per pagare la transazione nota come “GAS”, l’indirizzo del blocco in cui mi trovo e del blocco di destinazione della transazione.

## MANUALE WEB APP

Di seguito sono esplicitate tutte le funzionalità messe a disposizione dall’applicazione sviluppata, denominata ‘CyberApp’.

* *LOGIN (screen pagina login)*

Come ogni applicazione che si rispetti, si può accedere alla home page solo a seguito di un login nel quale è necessario inserire delle credenziali: l’e-mail e la password. Queste ultime sono sottoposte ad una serie di controlli che si occupano di confrontare i dati inseriti nella text-label della pagina di login con i dati presenti nel database di MongoDB relativo agli utenti; solo se vi è una congruenza verrà effettuato l’accesso, in caso contrario si visualizzerà un messaggio di errore esplicativo.

Una volta loggato come admin, è possibile registrare nuovi utenti passando alla schermata *Register* che si occupa di registrare tutti i dati dell’utente, compresa la mail e la password, che permettono poi di potersi loggare. In questa fase è stata utilizzata la libreria ‘bcrypt’ che cripta la password in fase di registrazione facendone l’hash e rendendola così sicura.

* *HOME (screen pagina home)*

Nella home page della nostra web-app saranno raffigurate, posizionate a griglia, le immagini caricate recentemente con le relative didascalie descrittive. Inoltre, è presente una barra di navigazione contenenti le seguenti opzioni: *User*, *Register*, *Images*, *Add Image e Logout*

* *SEZIONE USER*

La sezione messa a disposizione sotto il nome di ‘User’ permette di effettuare una ricerca sugli utenti memorizzati nel database relativo agli utenti di MongoDB. Per utenti intendiamo tutti i dipendenti dell’impresa edilizia che lavorano per il soddisfacimento degli obiettivi del progetto. Inoltre, in caso di necessità, dà la possibilità di eliminare un utente dalla lista degli user registrati.

* *SEZIONE REGISTER*

Il segmento ‘Register’ consente di inserire un nuovo utente, o meglio un nuovo dipendente nella lista dei lavoratori che collaborano al progetto. In questa pagina è consentito registrare un user all’interno del database inserendo *Name*, *Surname*, *Email* e *Password*. In questo caso la password, come detto precedentemente, viene criptata e nascosta grazie all’implementazione di metodi della libreria ‘bcrypt’.

Nel caso in cui si presentino degli errori, come la registrazione di un utente già presente del database, apparirà a video un messaggio di errore che aiuta il direttore dei lavori ad individuare l’errore ed a correggerlo.

* *SEZIONE IMAGES*

Nella sezione ‘Image’ è possibile ricercare le immagini presenti all’interno del nostro database MongoDB inserendo esclusivamente il nome associato all’immagine che stiamo ricercando e la data in cui essa è stata caricata. Nel momento della visualizzazione verrà visualizzata l’immagine, il titolo descrittivo a lei associata ed i metadati ad essa riferiti.

Inoltre, si ha l’opportunità, tramite dei button che portano ad una nuova schermata, di selezionare un’immagine caricata e modificarla od eliminarla.

* *SEZIONE ADD IMAGE*

Tale sezione permette di inserire nel nostro database delle immagini una nuova foto scattata dal drone. Il caricamento della nuova immagine è fattibile inserendo il nome descrittivo dell’immagine, la data in cui essa viene caricata, la data in cui essa è stata generata, una lieve descrizione e, naturalmente, la copia dell’immagine stessa.

Anche in questo caso, come nel caso dell’inserimento di un utente, in caso di problemi nella registrazione di una nuova immagine verrà stampato a video un messaggio di errore.

## PROGRESSI FUTURI

I due applicativi realizzati lavorano separatamente, inquanto la loro integrazione richiedono conoscenze che vanno al di là delle nozioni offerte da questo corso. Un futuro aggiornamento di questa quindi riguarda sicuramente il merge delle due app, andando a semplificare le azioni da eseguire per l’aggiunta di informazioni sul giornale dei lavori, avendo un solo software.

Un altro possibile upgrade riguarda la parte di log in: si può prevedere la differenziazione degli utenti a seconda del ruolo che ricopre, e pertanto anche delle operazioni consentite. Il campo del ruolo è già stato aggiunto nel modello dell’utente, nel momento della registrazione di un nuovo utente il campo ruolo viene impostato come “employee”, mentre quello del direttore dei lavori di default ad “admin”. Quello che manca è dunque la differenziazione delle azioni possibili e schermate visualizzabili, implementabili utilizzando il campo *“ruolo”* come discriminante.