

Progetto Ingegneria dei Requisiti
"IoT Sensors for Smart Tracking"

Mattia Bernardi

VR504232

Davide Bianchini

VR509340

Ilaria Meghi

VR516348

22 settembre 2025

Indice

| | | |
|----------|--|-----------|
| 1 | Introduzione | 3 |
| 1.1 | Descrizione del progetto | 3 |
| 1.2 | Funzionamento generale del sistema | 3 |
| 1.3 | Obiettivi del sistema | 3 |
| 2 | Glossario | 4 |
| 2.1 | Termini principali | 4 |
| 3 | Diagram | 6 |
| 3.1 | Intentional view | 6 |
| 3.1.1 | Data Retrieved | 7 |
| 3.1.2 | Data Transmitted | 7 |
| 3.1.3 | Data Stored | 8 |
| 3.1.4 | Dashboard | 9 |
| 3.2 | Structural view | 10 |
| 3.3 | Responsibility view | 11 |
| 3.3.1 | Agent Responsibility | 11 |
| 3.3.2 | Dependency Diagram | 12 |
| 3.4 | Functional view | 14 |
| 3.5 | Behavioural view | 16 |
| 4 | Conclusione | 18 |
| 4.1 | Elicitazione dei requisiti | 18 |
| 4.2 | Lista delle sorgenti utilizzate | 18 |

1 Introduzione

1.1 Descrizione del progetto

Il progetto riguarda lo sviluppo di una soluzione IoT pensata per il monitoraggio delle condizioni ambientali di beni sensibili durante il loro trasporto e stoccaggio.

Prodotti come: medicinali, opere d'arte, vino, alimenti freschi o materiali delicati richiedono una particolare attenzione, poiché parametri quali: temperatura, umidità, esposizione alla luce e vibrazioni devono rimanere entro certe soglie prestabilite, in quanto il superamento di tali valori comprometterebbe in modo irreversibile la qualità del prodotto.

1.2 Funzionamento generale del sistema

Il sistema utilizza sensori IoT, collocati all'interno dei pacchi, i quali rilevano periodicamente i parametri ambientali e la loro posizione. Successivamente, questi dati vengono trasmessi ad un'infrastruttura cloud dove vengono organizzati, cifrati e archiviati in modo sicuro.

Infine, una dashboard dedicata permette ai proprietari dei dispositivi di accedere in tempo reale alle informazioni e ricevere eventuali allarmi in caso di superamento delle soglie ammesse.

1.3 Obiettivi del sistema

Il sistema è stato progettato perseguiendo i seguenti obiettivi principali:

- **Trasparenza e fiducia:** garantire la possibilità di consultare in autonomia le condizioni della merce. Fornire tracciabilità certificata, grazie all'utilizzo di una blockchain, a garanzia della qualità dei beni monitorati.
- **Economicità:** utilizzare hardware a basso costo, riutilizzabile in più scenari applicativi, grazie a personalizzazione software.
- **Affidabilità e sicurezza:** assicurare la protezione dei dati lungo l'intero processo tramite autenticazione dei dispositivi e trasmissione sicura.
- **Scalabilità:** adottare un'architettura cloud, in grado di estendersi senza richiedere infrastrutture ridondanti.

2 Glossario

2.1 Termini principali

| Componente | Descrizione |
|-----------------------------|--|
| Device | Componente hardware che contiene i sensori e permette la raccolta e la trasmissione dei dati ambientali. |
| Sensor | Sensori di temperatura, umidità, vibrazioni, luce e posizione installati nel device. |
| DeviceOwner | Soggetto che possiede il device e ha accesso ai dati raccolti. |
| SoftwareRoutine | Monitoraggio integrità hardware e software del device. |
| Data | Insieme di valori misurati dai sensori. |
| DataProcessor | Modulo che elabora i dati grezzi. |
| DataStructured | Stato in cui i dati sono stati organizzati in un formato coerente. |
| DataEncryptionModule | Componente che cifra i dati per garantire sicurezza. |
| DataEncrypted | Stato in cui i dati sono stati criptati. |
| ConnectionModule | Modulo responsabile della gestione delle connessioni di rete. |
| DataLoggerModule | Modulo che registra e conserva i dati. |
| FallBackDataLogger | Modalità di salvataggio locale dei dati quando la trasmissione fallisce. |
| CorruptedData | Stato in cui i dati risultano danneggiati o non più affidabili. |
| UncorruptedData | Dati integri e non alterati. |
| BlockchainBasedStorage | Archiviazione che sfrutta la blockchain per garantire immutabilità e tracciabilità dei dati. |
| SmartContracts | Programmi automatici registrati su blockchain che gestiscono regole e processi in modo sicuro. |
| AccessControlSystem | Sistema che regola chi può accedere ai dati e con quali permessi. |
| AuthenticationSystem | Meccanismo che verifica l'identità degli utenti prima di consentire l'accesso. |
| NetworkTrafficControlSystem | Sistema che monitora e regola il traffico di rete per prevenire intrusioni. |

| Componente | Descrizione |
|-----------------------------|--|
| NextGenerationFirewall | Firewall avanzato che combina filtraggio pacchetti, ispezione applicativa e controllo delle minacce. |
| MFA Login | Autenticazione multifattore per aumentare la sicurezza degli accessi. |
| DataRecoveredFromBackup | Ripristino dei dati tramite copie di backup. |
| BackupSystem | Infrastruttura per la creazione e gestione di copie di sicurezza dei dati. |
| ETL Tool | Strumento che esegue le fasi di Extract, Transform e Load (estrazione, trasformazione e caricamento dei dati). |
| EfficientDashboard | Obiettivo di avere una dashboard efficiente. |
| DevelopmentTeam | Team responsabile dello sviluppo della dashboard. |
| UIDesigner | Figura che progetta l'interfaccia utente. |
| FrontendDeveloper | Sviluppatore lato client che implementa l'interfaccia. |
| ResponsiveDesign | Design adattivo per vari schermi (PC, tablet, smartphone). |
| DefinedAlertThreshold | Soglia di allerta per il superamento delle condizioni ambientali. |
| CustomizableAlertThresholds | Possibilità per l'utente di personalizzare le soglie di allerta. |
| AutomatedAlertsSystem | Sistema che gestisce automaticamente gli avvisi. |
| AlertTriggeringRoutine | Routine che gestisce il trigger degli avvisi. |
| RetryMechanism | Meccanismo di ritentativo in caso di fallimento. |
| PushSMSFailover | Invio SMS come backup in caso di fallimento della notifica principale. |

3 Diagram

3.1 Intentional view

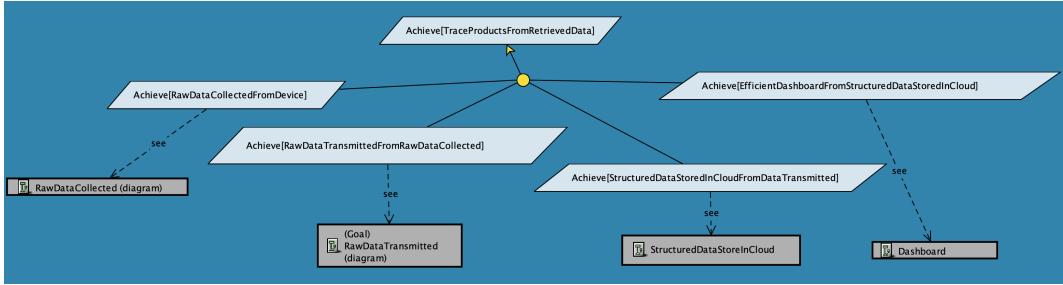


Figura 1: Goal Diagram

Il diagramma rappresenta il funzionamento generale del sistema, il cui obiettivo principale è tracciare i prodotti attraverso i dati raccolti durante il trasporto e lo stoccaggio. Per raggiungere questo scopo, il sistema deve completare quattro milestones fondamentali:

1. **Achieve[RawDataCollectedFromDevice]**: Raccolta dei dati grezzi attraverso l'utilizzo dei sensori.
2. **Achieve[RawDataTransmittedFromRawDataCollected]**: Trasmissione dei dati raccolti dai dispositivi verso il cloud.
3. **Achieve[StructuredDataStoredInCloudFromDataTransmitted]**: Organizzazione e archiviazione dei dati nel cloud.
4. **Achieve[EfficientDashboardFromStructuredDataStoredInCloud]**: Realizzazione di una dashboard efficiente e accessibile per analisi e monitoraggio.

In sintesi, il sistema ha un flusso completo: raccoglie → trasmette → organizza/archivia → visualizza i dati. In questo modo, garantisce il tracciamento affidabile della merce.

3.1.1 Data Retrieved

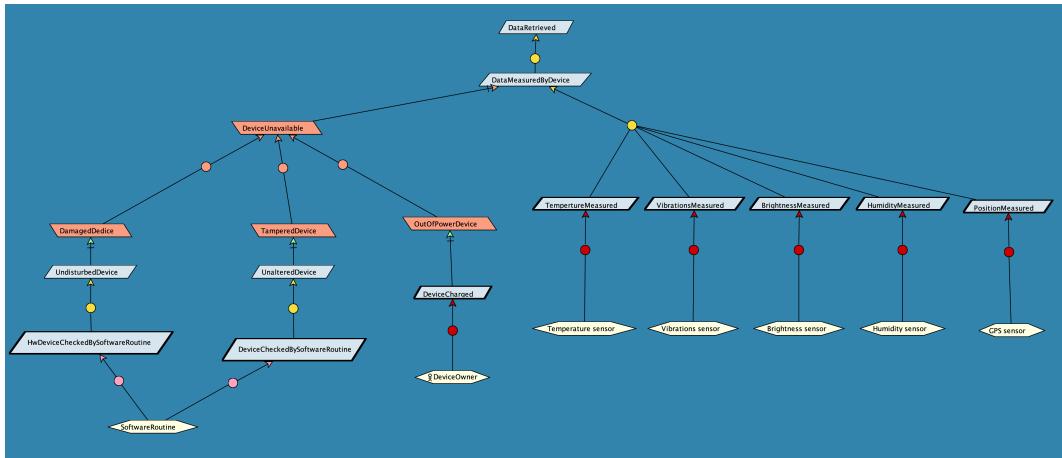


Figura 2: Data Retrieved

Questo diagramma rappresenta la fase di raccolta dei dati.

Il device è composto da sensori di temperatura, vibrazioni, luminosità, umidità e posizione, che permettono di rilevare i parametri da monitorare.

Analizzando questa fase, sono stati individuati possibili ostacoli ai goal, come problemi hardware o manomissioni software del dispositivo. La routine software si occupa di effettuare dei controlli per verificare che tutto funzioni correttamente.

3.1.2 Data Transmitted

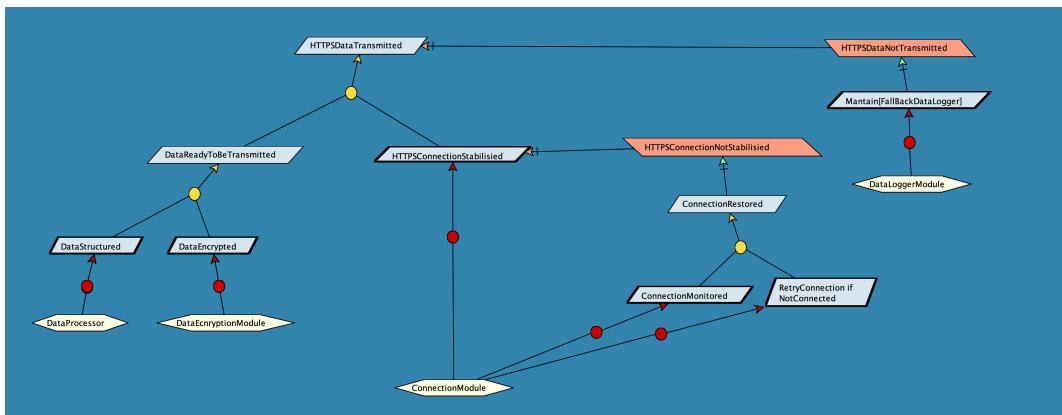


Figura 3: Data Transmitted

I dati, periodicamente raccolti, vengono inviati al cloud tramite il Connection Module, che instaura e mantiene una connessione HTTPS sicura e gestisce eventuali tentativi di riconnessione in caso di interruzioni. I dati vengono inviati dopo essere stati:

- **Strutturati [DataStructured]**: per evitare la trasmissione di dati nulli o corrotti.
- **Crittografati [DataEncrypted]**: per garantirne la sicurezza.

Nel caso in cui la trasmissione fallisca, è previsto un processo di fallback tramite data logger, attraverso il quale i dati vengono memorizzati localmente e successivamente inviati, garantendo così la disponibilità delle informazioni.

3.1.3 Data Stored

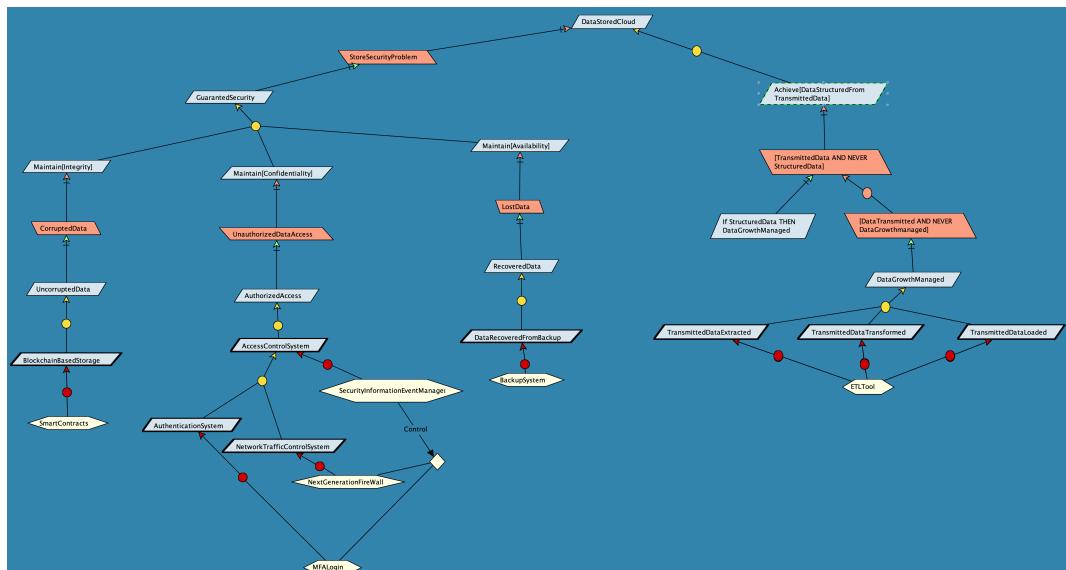


Figura 4: Data Stored

Una volta trasmessi, i dati vengono sottoposti ad un leggero processo di ETL per normalizzarli e prepararli al passo successivo, ovvero il salvataggio sicuro sul cloud. Per raggiungere questo obiettivo e per garantire integrità, confidenzialità e disponibilità, sono stati modellati dei maintain goal specifici.

In particolare, l'uso della blockchain consente di certificare la filiera del prodotto, assicurando che i dati misurati dai device non siano stati manomessi.

La confidenzialità dei dati è tutelata attraverso meccanismi di Login, sistemi ACS

e di controllo del traffico, che impediscono accessi non autorizzati. Infine, sistemi di backup dedicati consentono di affrontare eventuali perdite di informazioni garantendo una continua disponibilità dei dati.

3.1.4 Dashboard

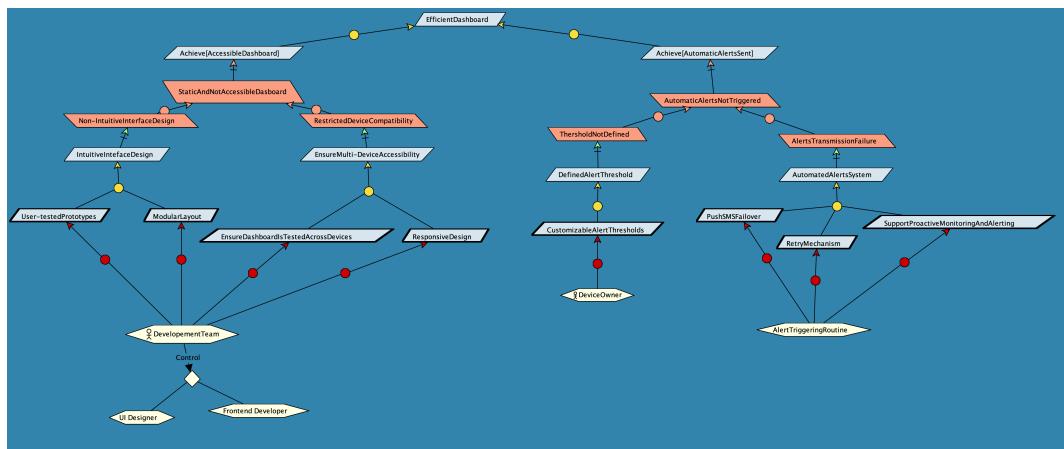


Figura 5: Dashboard

I dati raccolti sono facilmente consultabili tramite una dashboard intuitiva, modulare e responsive.

Questa interfaccia consente di visualizzare in tempo reale i parametri misurati dai sensori (come temperatura, umidità, luminosità, vibrazioni e posizione), presentandoli in modo chiaro e immediato.

Oltre alla semplice consultazione, la dashboard offre funzionalità avanzate: il Device Owner può impostare soglie di monitoraggio personalizzate per ciascun parametro e ricevere notifiche automatiche qualora vengano superati i valori definiti. In questo modo, la dashboard non rappresenta soltanto uno strumento di analisi, ma diventa un vero e proprio centro di controllo che garantisce un monitoraggio continuo

3.2 Structural view

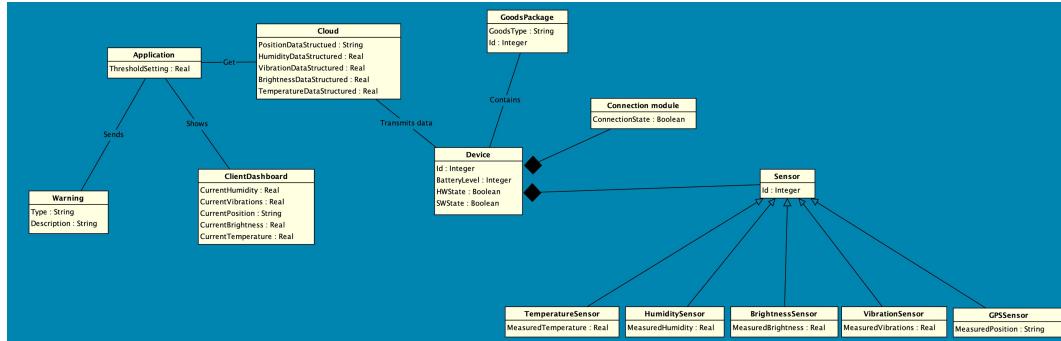


Figura 6: Conceptual Object

Il class diagram rappresenta l'architettura del sistema e descrive in modo strutturato le entità principali, insieme alle relazioni che intercorrono tra di esse.

Al centro troviamo la classe *Application*, la quale costituisce il core del sistema, in quanto permette al *Device Owner* di gestire la configurazione delle soglie critiche (*ThresholdSetting*).

L'applicazione, inoltre, recupera le informazioni dal *Cloud* (vedi relazione *Get*) e le mostra nella *dashboard*, modulo dell'applicazione (vedi relazione *Shows*). Infatti i dati acquisiti (attraverso i sensori) sono resi disponibili in tempo reale attraverso la *ClientDashboard*, che mostra i valori correnti dei parametri monitorati, permettendo così un controllo immediato delle condizioni della merce.

Infine, l'applicazione invia le notifiche di allerta ogni qual volta che un valore misurato supera le soglie configurate in precedenza; viene generato automaticamente un *Warning* (vedi relazione *Sends*).

Come spiegato in precedenza, *Device* rappresenta il dispositivo IoT fisico inserito all'interno dei pacchi. *Device* è composto da vari sensori che sono specializzazioni della classe *Sensor*.

3.3 Responsibility view

3.3.1 Agent Responsibility

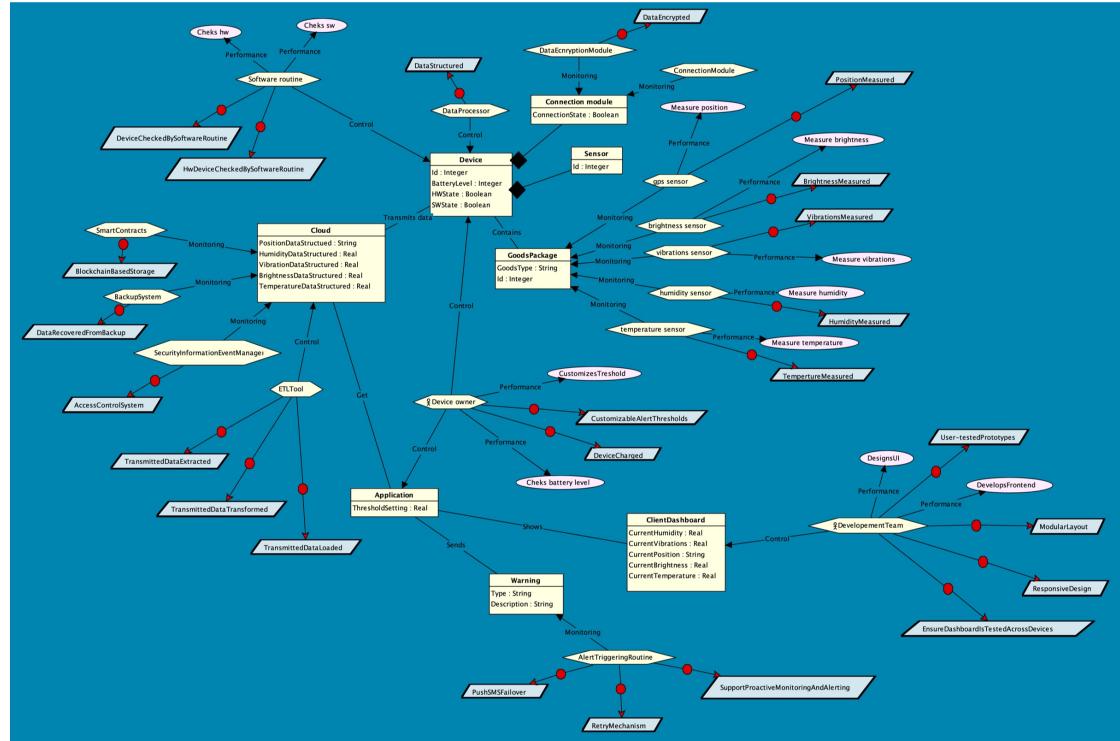


Figura 7: Agent Responsibility

L'agent diagram è un diagramma che ha come prospettiva principale gli agenti e permette di rappresentare le responsabilità, le capacità e le operazioni degli agenti. Il diagramma mostra come diversi agenti specializzati collaborano all'interno del sistema: alcuni raccolgono dati, altri li elaborano o li proteggono, altri ancora li presentano all'utente finale. Agenti principali:

- **DataProcessor**: responsabile dell'elaborazione dei dati.
- **SoftwareRoutine**: responsabile di effettuare dei controlli per verificare che il dispositivo non sia danneggiato e manomesso.
- **ConnectionModule**: gestisce lo stato della connessione e il monitoraggio del collegamento.
- **Sensors**: ciascuno è responsabile della misurazione di un parametro specifico (posizione, luminosità, vibrazioni, umidità, temperatura).

- **SecurityInformationEventManager**: responsabile della gestione degli eventi di sicurezza e dell'accesso ai dati.
- **DevelopmentTeam**: responsabile della realizzazione e del miglioramento del frontend, del design UI e della compatibilità tra dispositivi.
- **DeviceOwner**: responsabile della configurazione delle Alert Threshold e di monitorare il livello della batteria ed eventualmente ricaricare il device.

Ogni agente ha una responsabilità precisa: i sensori monitorano la merce, i moduli come il DataProcessor e il Connection module elaborano e trasmettono dati, mentre agenti come ETL Tool e Security Manager garantiscono sicurezza e integrità dei dati. Environment Agents come il DevelopmentTeam e il Device Owner hanno invece responsabilità legate all'usabilità del sistema e alla gestione delle alert thresholds .

3.3.2 Dependency Diagram

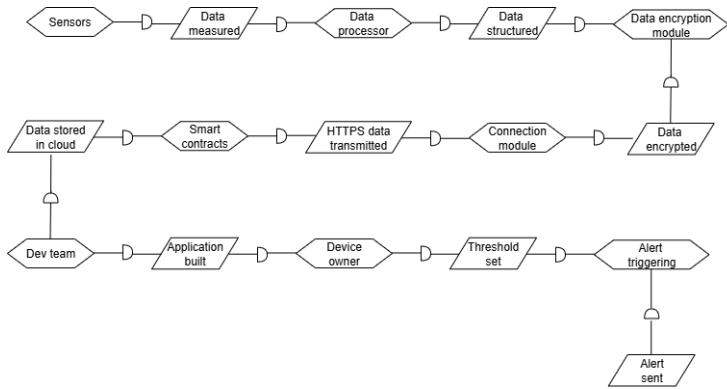


Figura 8: Dependency Diagram

Il dependency diagram rappresenta una mappa del flusso informativo del sistema, tracciando il ciclo informativo, dall'acquisizione dei dati fino al loro utilizzo finale. L'intero processo ha origine dai *Sensors*, che fungono da porta d'ingresso per tutti i dati del sistema, misurando periodicamente i dati grezzi dei parametri ambientali (temperatura, umidità, luminosità, vibrazioni e posizione) e generando quello che nel diagramma viene identificato come *Data measured*.

Successivamente, il *Data processor* assume il ruolo di primo elaboratore, ricevendo questo flusso di misurazioni e trasformandole in *Data structured*, ovvero informazioni organizzate secondo un formato standardizzato che ottimizza tutte le operazioni successive di analisi e gestione.

Una volta strutturati, il *Data encryption module* crittografa i dati, trasformandoli in *Data encrypted*, in modo tale da proteggere le informazioni sensibili da accessi non autorizzati durante la fase di trasmissione e archiviazione. Il Connection module, poi, si occuperà di implementare protocolli di comunicazione sicuri che garantiscono l'integrità e l'autenticità dei dati durante il trasferimento attraverso la rete.

Una volta stabilita una connessione sicura, gli *smart contract* applicano automaticamente le regole di business e i protocolli di gestione dei beni monitorati. Grazie alla blockchain, i dati vengono registrati in modo immutabile, creando una copia sicura e non modificabile che garantisce la tracciabilità del percorso della merce. Successivamente, tali dati vengono archiviati nel *cloud*, dove costituiscono la base informativa per tutte le operazioni successive.

Il team di sviluppo realizza poi l'applicazione, dotata di una dashboard che consente agli stakeholders di visualizzare in modo semplice ed elegante le informazioni archiviate nel cloud. Questa dashboard non è solo uno strumento di consultazione, ma rappresenta anche il punto di partenza per le operazioni successive, come la possibilità per i Device owner di impostare soglie personalizzate sui parametri monitorati.

Infatti, la parte inferiore del diagramma mostra il sistema di gestione degli alert. Qui il processo parte dal *Device owner*, che imposta le soglie di attenzione per i parametri ambientali da monitorare. Queste soglie vengono poi inviate al componente di *Alert triggering*, che confronta continuamente i valori reali con i limiti definiti. Se i parametri superano i valori consentiti, viene generato un *alert* e inviato automaticamente agli stakeholders (via SMS, email o notifiche push). In questo modo, chi di dovere viene subito informato e può intervenire rapidamente.

È quindi evidente che, pur trattandosi di un processo continuo, tutte le fasi dipendono dal dato iniziale. La misurazione in tempo reale rappresenta infatti il punto di partenza imprescindibile: senza di essa non sarebbe possibile attivare la trasmissione, l'archiviazione né, in ultima istanza, la rappresentazione dei dati stessi.

3.4 Functional view

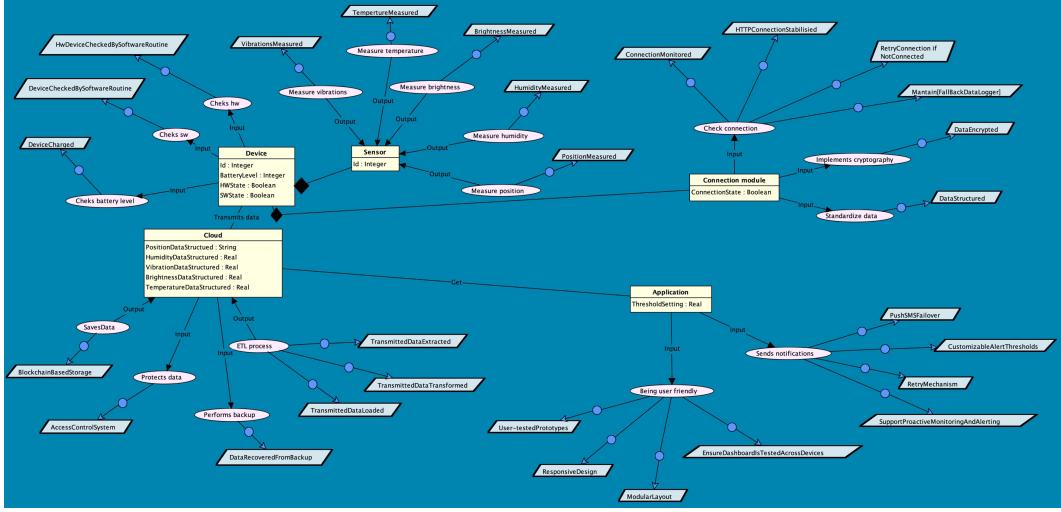


Figura 9: Operationalization Diagram

L’operationalization diagram si focalizza sulle operazioni necessarie a soddisfare i goal, mostrando come le operazioni modificano lo stato del sistema sfruttando degli input e producendo degli output. Tra le operazioni chiave:

- **Check hw/sw**: Controlli periodici software e hardware per garantire l’integrità del sistema e il corretto funzionamento di tutti i componenti.
- **Checks battery level**: Monitoraggio continuo dell’autonomia energetica del dispositivo per garantire la continuità operativa.
- **Measure temperature/humidity/brightness/vibrations/position**: Acquisizione di parametri ambientali critici attraverso sensori specializzati.
- **Standardize data**: Normalizzazione dei dati provenienti da diverse sorgenti sensoriali per garantire uniformità nell’elaborazione.
- **Implements cryptography**: Protezione dei dati sensibili attraverso meccanismi crittografici avanzati per la sicurezza end-to-end.
- **Check connection**: Verifica continua della stabilità del collegamento di rete con monitoraggio dello stato della connessione.
- **ETL process**: Gestione del flusso sicuro dei dati verso il cloud attraverso operazioni di Extract, Transform, Load con controlli di integrità.

- **Sends notifications:** Sistema di alerting multi-canale per notifiche in tempo reale basate su soglie personalizzabili.
- **Performs backup:** Meccanismi di persistenza e recuperabilità dei dati critici.
- **Protect data:** Gestione dell'autorizzazione e autenticazione degli utenti con controllo granulare dei permessi.

3.5 Behavioural view

I sequence diagram mostrano l'interazione temporale tra i diversi componenti del sistema, evidenziando il flusso di messaggi e le operazioni eseguite in sequenza per realizzare le funzionalità del sistema. Il primo diagramma illustra il flusso completo dalla verifica dello stato del dispositivo fino alla trasmissione dei dati al cloud:

- **Device Status Check:** Controllo periodico dello stato del dispositivo con gestione di scenari alternativi (damaged, tampered, out of power) e relative azioni correttive.
- **Data Acquisition:** Acquisizione di tutti i parametri sensoriali (temperatura, vibrazioni, luminosità, umidità, posizione) con strutturazione e crittografia dei dati.
- **Data Transmission:** Stabilimento della connessione HTTPS e trasmissione dei dati con gestione dei fallimenti di connessione e meccanismi di retry automatici.

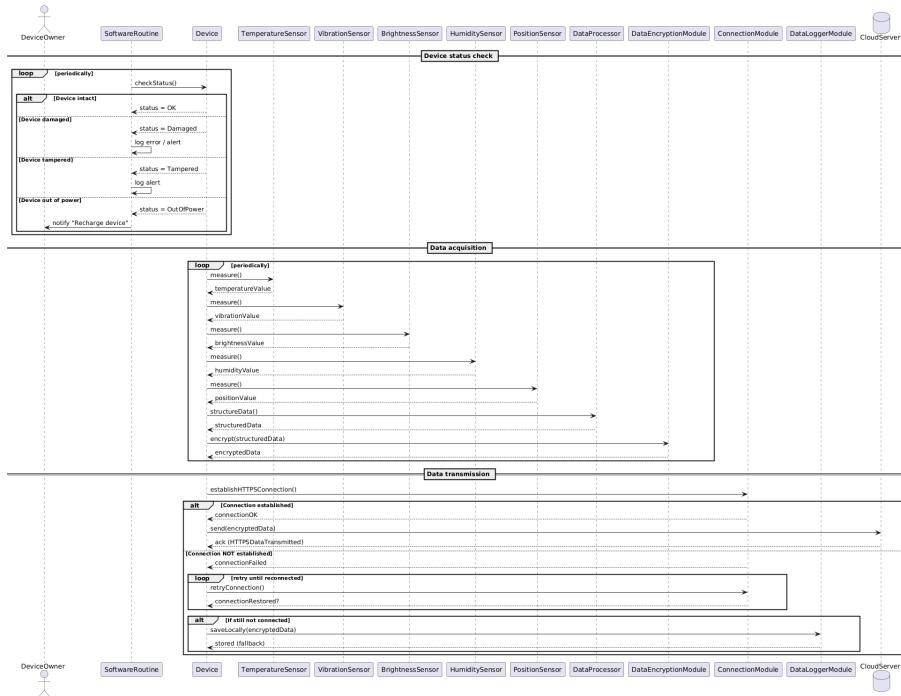


Figura 10: Sequence Diagram

Il secondo diagramma mostra la gestione dei dati nel cloud e il sistema di notifiche:

- **Cloud Data Management:** Archiviazione dei dati crittografati con gestione del backup e recovery in caso di corruzione, oltre al controllo degli accessi non autorizzati.
- **Dashboard and Threshold Management:** Aggiornamento dei dati strutturati, visualizzazione tramite dashboard e gestione delle soglie personalizzabili con sistema di alerting.
- **Multi-channel Notification:** Sistema di notifiche che include alerting primario e SMS failover per garantire la consegna delle notifiche critiche.

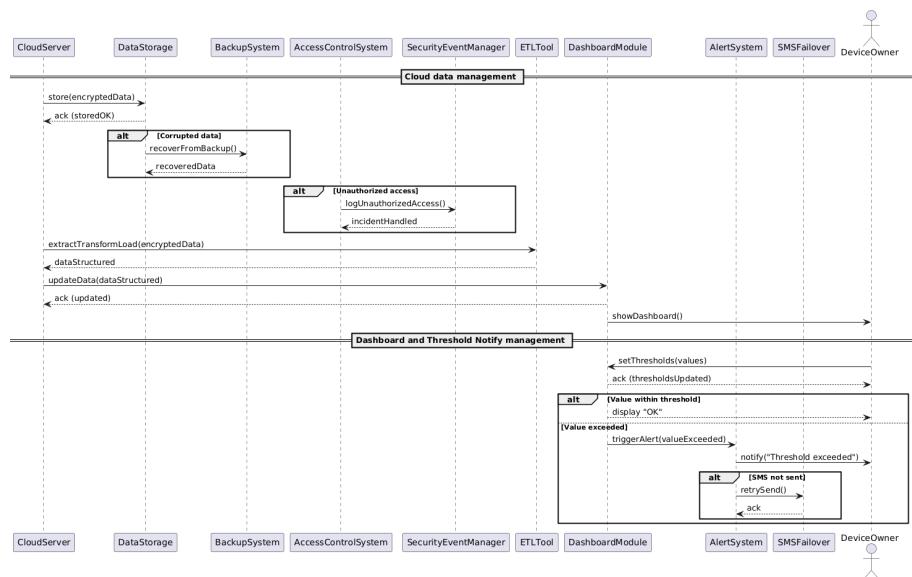


Figura 11: Sequence Diagram 2

4 Conclusione

4.1 Elicitazione dei requisiti

Abbiamo svolto il lavoro principale in gruppo, ponendoci nei panni degli stakeholder. La fase di elicitazione dei requisiti è stata realizzata principalmente attraverso **interviste e sessioni di gruppo**. Grazie alle interviste abbiamo identificato gli obiettivi principali del sistema, le funzionalità da implementare, i requisiti operativi da rispettare e le aspettative quali facilità d'uso, efficienza e possibilità di personalizzazione. Successivamente, le sessioni di gruppo hanno avuto un ruolo centrale: si sono svolte nell'arco di più giorni, con incontri finalizzati a documentare e approfondire i risultati emersi. Le sessioni non strutturate, come il brainstorming, hanno permesso di portare alla luce aspetti nascosti del sistema, esplorare problemi e soluzioni in maniera più ampia e stimolare la collaborazione e la creatività del gruppo.

4.2 Lista delle sorgenti utilizzate

Per sviluppare il progetto ci siamo basati su diverse fonti, come articoli, blog specializzati e siti di divulgazione tecnica. Abbiamo usato motori di ricerca per capire meglio il contesto e raccogliere idee. Le risorse consultate ci hanno aiutato a comprendere le tecnologie IoT, il tracciamento intelligente e la logistica, fornendo spunti pratici e concettuali per realizzare il nostro lavoro. Tra le principali fonti utilizzate ci sono:

- IBM – Internet of Things: cos’è e perché è importante
<https://www.ibm.com/think/topics/internet-of-things>
- Towards an IoT and Blockchain-based System for Monitoring and Tracking Agricultural Products
https://www.researchgate.net/publication/372611061_Towards_an_IoT_and_Blockchain-based_System_for_Monitoring_and_Tracking_Agricultural_Products
- How IoT Drives Cold Chain Logistics
<https://www.mokosmart.com/how-iot-drives-cold-chain-logistics/>
- Mastering IoT Testing: Strategies and Tools
<https://luxequality.com/blog/mastering-iot-testing-strategies-tools/>

Queste fonti ci hanno dato idee pratiche e ispirazione per progettare un sistema funzionale e moderno.