Verifica di SISTEMI E RETI classe 5^A ROB.

# Progettare smart mobility.

Il comune di Cuneo vuole rinnovare l’intera infrastruttura di gestione della mobilità urbana utilizzando tecnologie più moderne. La nuova infrastruttura sarà composta dai seguenti dispositivi **IoT**:

1. **semafori** in tutti gli incroci stradali, controllati da un server centrale dedicato,
2. sensori per la **misura del traffico** veicolare sulle principali arterie della città,
3. **telecamere** utili per il rilevamento di accessi in zone ad accesso limitato o infrazioni quali invasione corsia dei mezzi pubblici o invasione della pista ciclabile. Ogni telecamera, previa elaborazione delle immagini su server apposito, permette il riconoscimento della targa dei veicoli. Le targhe sono utili per sanzionare violazioni, o individuare veicoli autorizzati.

Inoltre l’infrastruttura sarà corredata di:

1. **server per il controllo dei semafori e l’elaborazione dei sensori di traffico**. In caso di fault del server, il sistema semaforico continuerà a funzionare autonomamente in modalità “non-IoT”.
2. **server per l’elaborazione delle immagini** e riconoscimento targhe dei veicoli.

Tutti i dati devono circolare all’interno della infrastruttura in **tempo reale** ed inoltre devono essere **registrati** sia per finalità amministrative che per il monitoraggio del buon funzionamento del sistema. Inoltre deve essere prevista una **interfaccia di controllo user-friendly** per lo smart mobility manager e per gli operatori che vogliano controllare, manutenere o configurare il sistema in tempo reale.

Analizza la realtà di riferimento e, fatte le opportune ipotesi aggiuntive, individua una soluzione che a tuo motivato giudizio sia la più idonea per sviluppare il progetto, anche mediante rappresentazioni grafiche, dell’infrastruttura tecnologica ed informatica necessaria a gestire il sistema nel suo complesso, dettagliando:

1. l’infrastruttura di comunicazione, in termini di caratteristiche dei canali, degli apparati e dei protocolli, che permetta di trasmettere e ricevere tutte le informazioni coinvolte;
2. le caratteristiche generali dei componenti hardware e software del sistema sia a livello centrale che nei dispositivi IoT;
3. le misure e gli apparati per assicurare la continuità dei servizi,
4. le misure necessarie onde non incorrere in violazioni della normativa GDPR.

**NOTA: descrivere per esteso tutte le ipotesi aggiuntive!**

**Consegnare un SOLO file su Classroom.**

**Spiegazione infrastruttura di rete:**

Nello schema sono rappresentati 2 esempi di impianti di sensori e semafori e il server centrale.

1. Nell’esempio numero 1 ho ipotizzato di trovarmi in un incrocio di dimensione variabile (accesso a 1, 2 o 3 corsie), per ogni carreggiata ci sarà una telecamera che supervisionerà il comportamento degli utenti della strada ed una ulteriore che riprenderà il centro dell’incrocio. Il tutto sarà gestito da 1 raspberry pi.
2. Nel secondo esempio viene rappresentato un controllo su una strada normale (ipotizzo di mettere uno di questi sistemi ogni 500 metri). Questi sistemi sono composti da 2 telecamere (una per che riprende a destra e una che riprende a sinistra) e un sensore di velocità che avrà anche la funzione di conteggio dei veicoli se necessario. Anche questo sistema sarà gestito da un Raspberry Pi e il tutto sarà collocato ad un lampione della luce pubblica in modo tale da usufruire dell’elettricità. Ho deciso di creare questo sistema compatto (tutto su un unico lampione e non separati) per agevolare i costi riguardanti Raspberry, lavori di installazione, la manutenzione fisica.

I dati ottenuti da questi sistemi verranno processati sul raspberry tramite un programma che si basa sull’IA per identificare le infrazioni (dalla telecamera principalmente) e successivamente solo i più rilevanti quali invasione di corsia, incidenti stradali, superamento del limite di velocità etc… saranno inviati al server centrale per evitare un carico eccessivo di dati sulla rete 4G. I Raspberry Pi avranno la possibilità di archiviare fino a 3 ore di video dopo di che il sistema sovrascriverà i filmati a partire da quelli più vecchi.

I sensori(telecamere e sensori di velocità) sono alimentati tramite PoE (Power of Ethernet) mentre i semafori sono alimentati tramite PoE ma anche tramite sistemi aggiuntivi. E in caso di mancato accesso internet dal Raspberry i semafori continueranno a dirigere il traffico tramite dati di Default presenti sul dispositivo.

Per la gestione dei sensori l’unico accesso presente è situato all’interno del comune di Cuneo, in una stanza chiusa con accesso limitato a poche persone per evitare che un malintenzionato possa configurare i sensori in maniera maligna. Il programmatore inserirà quindi i dati che verranno inviati ai vari sensori (linea tratteggiata viola).

Nello stesso comune ci sarà un programma dedicato all’elaborazione delle immagini che tramite un’IA estrae la targa

dei veicoli e a seconda dell’infrazione (invasione di corsia, infrazione limite stradale, precedenze) vengono categorizzati in un database. Le infrazioni di grave entità (incidente stradale) verranno inserite in una ulteriore tabella che dovrà essere successivamente analizzata da un addetto alla sicurezza.

Nel DB1 vengono inseriti i parametri con cui stanno lavorando i sensori e la cronologia con i dati relativi a chi ha aggiornato il sistema di funzionamento.

Nel DB2 vengono inseriti i dati relativi alle infrazioni come scritto in precedenza.

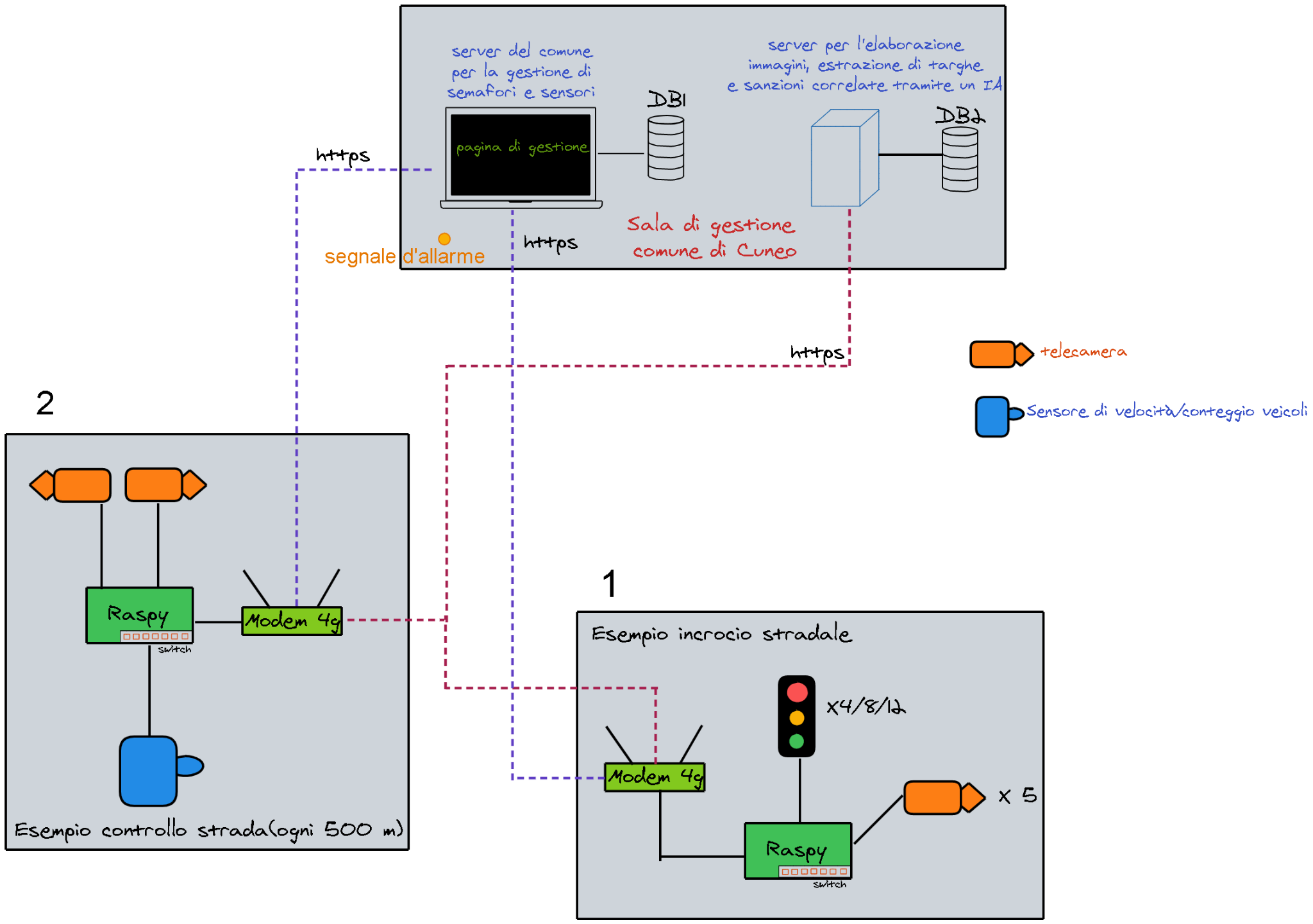
I problemi in cui si può incorrere sono:

problemi di alimentazione→ gruppo di continuità (principalmente per i semafori)

Problemi di cablaggio → si necessita di una manutenzione fisica

Problemi di sicurezza nel perimetro del server → accesso limitato tramite badge, o sistemi più elevati quali scansione biometrica e unica chiave fisica nelle mani del direttore.

Problemi fisici(non funzionamento di led semaforici)--> immediatamente il Raspberry inizializza il sistema di pericolo(luci gialle lampeggianti) e contatta il centro operativo tramite un alert o una spia luminosa.



Link schema: https://excalidraw.com/#json=6449175378001920,PkR\_w8FVewDQ0EEgMLUisw