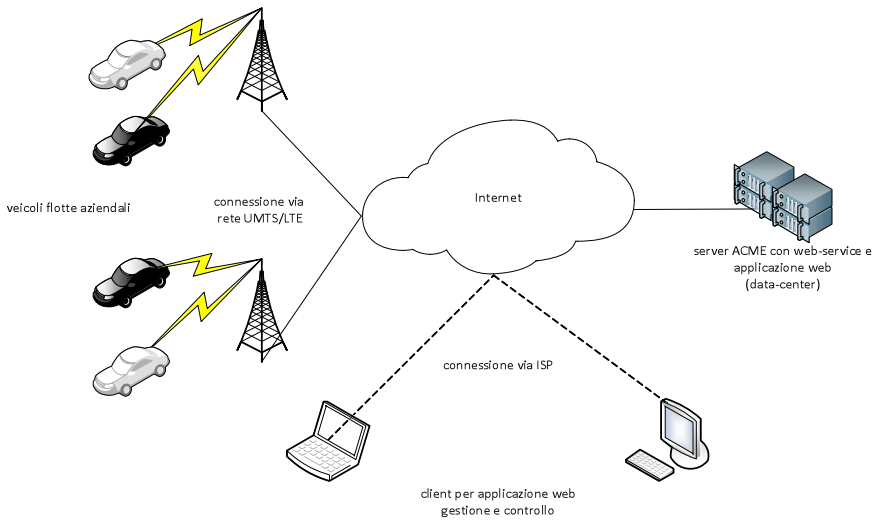
Traccia di soluzione della <u>seconda</u> simulazione della prova scritta di "Sistemi e reti" per l'indirizzo "Informatica e Telecomunicazioni" articolazione "Informatica" dell'Istituto tecnico settore tecnologico (MIUR 2016)

- ## Ipotesi aggiuntive

  Anche in considerazione della limitata quantità di dati che devono essere trasmessi/ricevuti dai dispositivi installati a bordo degli automezzi, si ritiene la connettività alla rete Internet offerta dalla rete telefonica cellulare con tecnologia UMTS/LTE adeguata per copertura e qualità alle esigenze del sistema di gestione e controllo delle flotte aziendali.

- ## Analisi della realtà e progetto del sistema

  Una tipica soluzione SaaS prevede un servizio ospitato su un server, fisico o virtuale, di un data-center a cui accedono i client che in questo caso sono sia i dispositivi installati a bordo degli automezzi, che i browser utilizzati per l'esecuzione dell'applicazione web per la gestione e il controllo delle flotte aziendali. I dispositivi installati a bordo degli automezzi eseguono una APP che accede al servizio invocando una web-API HTTP esposta da un web-service di tipo REST, mentre l'applicazione di gestione e controllo delle flotte aziendali è resa disponibile alle aziende clienti nella forma di un'applicazione web accessibile mediante un normale browser per pagine HTML/JavaScript. Il web-service e l'applicazione web interagiscono con un unico database gestito da un DBMS transazionale.



- ## Caratteristiche e funzionalità dei dispositivi installati a bordo degli automezzi

  I dispositivi installati a bordo degli automezzi di ciascuna flotta aziendale devono avere le seguenti caratteristiche:

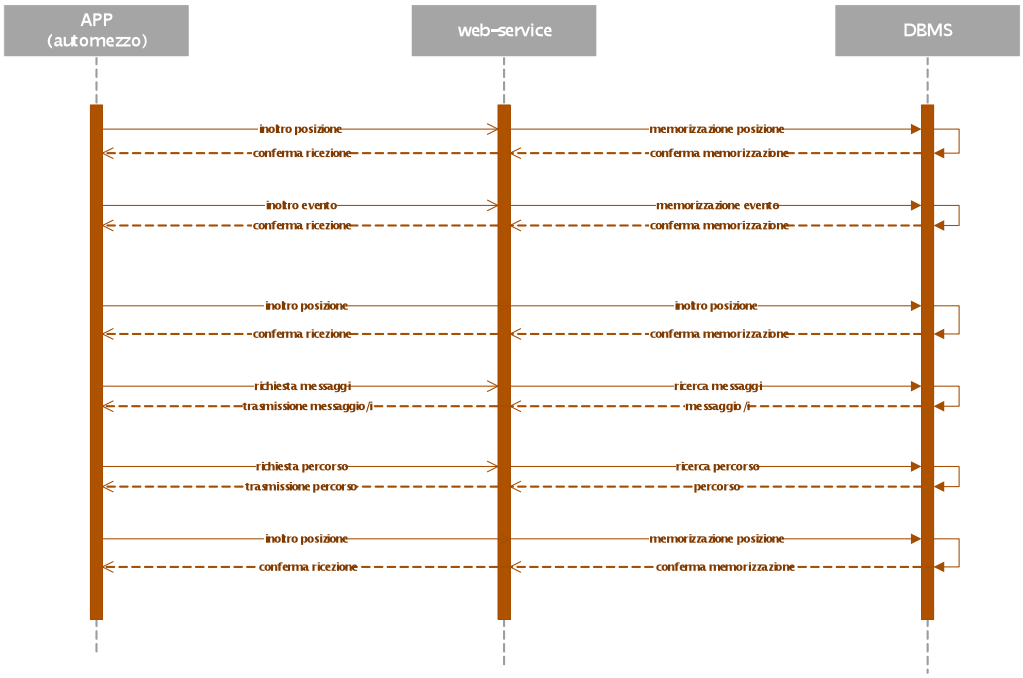
  - connettività UMTS/LTE per l'accesso continuativo alla rete Internet (eventualmente con antenna esterna)
  - dispositivo GPS integrato (eventualmente con antenna esterna)
  - display grafico a colori per la visualizzazione dell'interfaccia utente della APP del sistema di gestione flotte

- connessione al sistema audio di bordo per la riproduzione vocale di messaggi e avvisi

- touchscreen per l'interazione dell'utente con la APP del sistema di gestione flotte

- microfono per l'interazione vocale dell'utente con la APP del sistema di gestione flotte

Il dispositivo può essere derivato da un tablet commerciale le cui caratteristiche in termini di memoria FLASH/RAM e di CPU sono senz'altro adeguate per l'esecuzione della APP del sistema di gestione flotte.
L'interazione della APP eseguita dal dispositivo con il web-service esposto dal servizio è esemplificata dal seguente diagramma UML di sequenza:



La trasmissione di informazioni dal web-service verso il dispositivo installato a bordo dell'automezzo avviene in risposta ad una interrogazione periodicamente ripetuta da parte della APP in esecuzione sul dispositivo stesso; in alternativa potrebbero essere utilizzati dei web-socket per rendere disponibile un canale di comunicazione in modalità *push*.

- **Protocolli di comunicazione**

La sicurezza dei dati scambiati tra i dispositivi installati a bordo degli automezzi e i server che espongono il web-service può essere garantita dall'adozione dello standard HTTPS basato su protocollo TLS[1], mentre la riservatezza dei dati delle diverse aziende che utilizzano il sistema può essere garantita da ACME mediante il rilascio di API *key* univocamente associate alle singole aziende clienti e la registrazione di username/password per l'autenticazione e la diversificazione dei diversi ruoli all'interno della singola azienda.
Di seguito è definita le web-API del web-service che il sistema di gestione flotte espone verso le APP dei dispositivi installati a bordo degli automezzi (URL: https://car-ws.acme.com/) nell'ipotesi che i singoli veicoli siano identificati dalla propria targa.

[1] in **[2]** alle pagine 140 – 145 e in **[3]** alle pagine alle pagine 1224 – 1227 sono riportate le informazioni essenziali sui protocolli TLS e HTTPS, comprendenti la generazione dei certificati per il server e la loro conservazione presso una CA (*Certification Autority*)

- **inoltro della posizione del veicolo al server (da effettuarsi periodicamente)**

RICHIESTA

metodo: POST
URL: https://car-ws.acme.com/position/*targa*?id=*key*
corpo (XML):
```
<position>
    <timestamp>…</timestamp>
    <latitude>…</latitude>
    <longitude>…</longitude>
    <speed>…</speed>
</position>
```

RISPOSTA

codice di stato: 200 = posizione registrata
400 = errore richiesta (targa inesistente o chiave non autorizzata)
500 = errore server

ESEMPIO RICHIESTA

POST https://car-ws.acme.com/position/AB123XYZ?id=1234567890ABCDEF
```
<position>
    <timestamp>2016-04-20T09:48:52</timestamp>
    <latitude>43.55</latitude>
    <longitude>10.3167</longitude>
    <speed>55.5</speed>
</position>
```

- **inoltro di un evento anomalo al server**

RICHIESTA

metodo: POST
URL: https://car-ws.acme.com/event/*targa*?id=*key*
corpo (XML):
```
<event>
    <timestamp>…</timestamp>
    <code>…</code>
    <description>…</description>
</event>
```

RISPOSTA

codice di stato: 200 = evento registrato
400 = errore richiesta (targa inesistente o chiave non autorizzata)
500 = errore server

ESEMPIO RICHIESTA

POST https://car-ws.acme.com/event/AB123XYZ?id=1234567890ABCDEF
```
<event>
    <timestamp>2016-04-20T09:48:52</timestamp>
    <code>123</code>
    <description>Guasto meccanico</description>
</event>
```

- **richiesta di un percorso impostato sul server (da effettuarsi periodicamente)**

RICHIESTA

metodo: GET
URL: https://car-ws.acme.com/route/*targa*?id=*key*

RISPOSTA

codice di stato: 200 = percorso fornito
400 = errore richiesta (targa inesistente o chiave non autorizzata)
500 = errore server

corpo (XML):
```
<route destination="…">
        <position>
                <latitude>…</latitude>
                <longitude>…</longitude>
        </position>
        <position>
                <latitude>…</latitude>
                <longitude>…</longitude>
        </position>
        …
        …
        …
</route>
```

ESEMPI RISPOSTA (IL SECONDO ESEMPIO È RELATIVO AD UN PERCORSO NON IMPOSTATO)

200
```
<route destination="Pisa">
    <position>
            <latitude>43.55</latitude>
            <longitude>10.3167</longitude>
    </position>
    <position>
            <latitude>43.6167</latitude>
            <longitude>10.4</longitude>
    </position>
</route>
```

200
```
<route destination="">
</route>
```

- **richiesta di una mappa specificando la posizione centrale e il raggio in Km**
(il server restituisce mappe a scala costante)

RICHIESTA

metodo: GET
URL: https://car-ws.acme.com/map?id=*key*
corpo (XML):
```
<map>
        <center>
                <latitude>…</latitude>
```



```
                    <longitude>…</longitude>
                </center>

                <radius>…</radius>
            </map>
```

RISPOSTA

codice di stato: 200 = mappa restituita
400 = errore richiesta (chiave non autorizzata o mappa non generabile)
500 = errore server

corpo (PNG): immagine

ESEMPIO RICHIESTA

GET https://car-ws.acme.com/map?id=1234567890ABCDEF

```
<map>
    <center>
        <latitude>43.55</latitude>
        <longitude>10.3167</longitude>
    </center>
    <ray>1.5</ray>
</map>
```

- **richiesta di uno o più messaggi impostati sul server (da effettuarsi periodicamente)**

RICHIESTA

metodo: GET
URL: https://car-ws.acme.com/text/*targa*?id=*key*

RISPOSTA

codice di stato: 200 = messaggio fornito
400 = errore richiesta (targa inesistente o chiave non autorizzata)
500 = errore server

corpo (XML):

```
<messages>
    <message>
        <timestamp>…</timestamp>
        <text>…</text>
    <message>
    <message>
        <timestamp>…</timestamp>
        <text>…</text>
    <message>
    …
    …
    …
</messages>
```

ESEMPI RISPOSTE (IL SECONDO ESEMPIO È RELATIVO ALL'ASSENZA DI MESSAGGI SUL SERVER)

200

```
<messages>
    <message>
        <timestamp>2016-04-20T09:48:52</timestamp>
        <text>Manutenzione prenotata per 22/04/2016</text>
    </message>
```

</messages>

200
<messages>
</messages>

- **richiesta di un messaggio vocale impostato sul server (da effettuarsi periodicamente)**

RICHIESTA

metodo:	GET
URL:	https://car-ws.acme.com/vocal/*targa*?id=*key*

RISPOSTA

codice di stato:	200 = messaggio fornito
	400 = errore richiesta (targa inesistente o chiave non autorizzata)
	500 = errore server
corpo (WAW):	audio

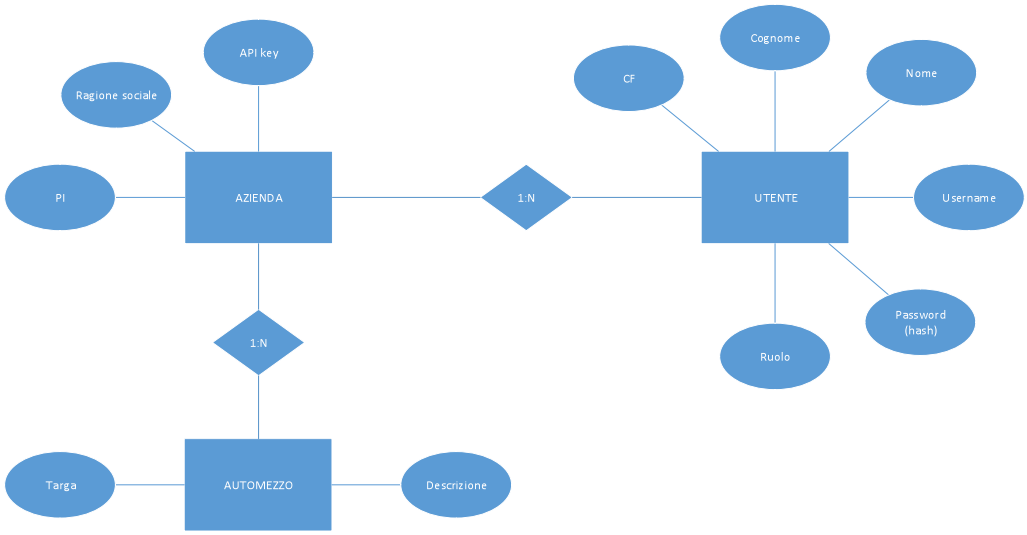
Quesiti

1)

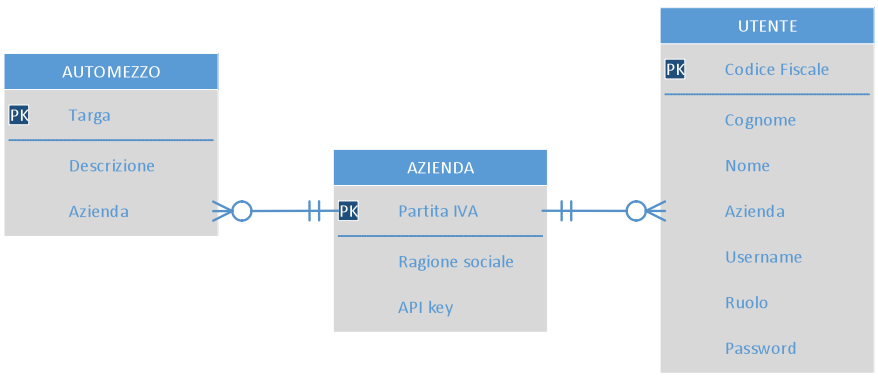
Nell'ipotesi che i ruoli di accesso ai dati memorizzati nel database del server ACME da parte di un utente di un'azienda siano

- autista (con privilegio di accesso in lettura ad un insieme limitato di informazioni)
- operatore (con privilegio di accesso in lettura/scrittura ad alcune informazioni quali posizioni di tutti i veicoli, eventi ricevuti dai veicoli, messaggi e percorsi inviati ai veicoli, …)
- amministratore (con privilegio in lettura/scrittura su tutte le informazioni, compresa la possibilità di modificare la flotta di veicoli, le credenziali degli autisti e degli operatori, …)

il seguente diagramma E/R modellizza la porzione di un database relazionale che permette di autenticare gli utenti del servizio – compresi gli automezzi in quanto la targa viene associata alla API *key* assegnata all'azienda – e di attribuire loro i corretti privilegi di accesso ai dati:



Tenendo conto che le password degli utenti – per motivi sia di sicurezza che di privacy – devono essere memorizzate in forma cifrata mediante una funzione *hash*, il database relazionale risultante può essere così schematizzato:



In **[1]** alle pagine 379 – 387 è riportato il codice in linguaggio PHP necessario per gestire l'autenticazione ed il controllo degli accessi degli utenti mediante l'uso delle sessioni e la generazione di specifiche pagine HTML: nel caso specifico deve essere integrata la verifica del privilegio dell'utente corrente in relazione alle operazioni che può effettuare.

2)

La web API HTTP del web-service esposto dal server ACME è documentata nella prima parte della soluzione; nell'ipotesi che i dispositivi installati a bordo degli automezzi siano dotati di S.O. Android il linguaggio di programmazione della APP è necessariamente Java; il codice che segue implementa una classe il cui costruttore inoltra al web-service la posizione e la velocità attuali dell'automezzo:

```
class SendPosition {
    private String url = "https://car-ws.acme.com/position/";
    private String body = "<?xml version=\"1.0\" encoding=\"UTF-8\"?>";
    private int status = 0;
    private int length;

    public SendPosition(String targa, String key, float latitude, float longitude, float speed) {
        try {
            // costruzione del timestamp nel formato ISO 8601
            TimeZone timezone = TimeZone.getTimeZone("UTC");
            DateFormat dateformat = new SimpleDateFormat("yyyy-MM-dd'T'HH:mmZ");
            dateformat.setTimeZone(timezone);
            String timestamp = dateformat.format(new Date());
            // costruzione del body XML da inoltrare al WS
            body += "<position>\r\n";
            body += "<timestamp>" + timestamp + "</timestamp>\r\n";
            body += "<latitude>" + Float.toString(latitude) + "</latitude>\r\n";
            body += "<longitude>" + Float.toString(longitude) + "</longitude>\r\n";
            body += "<speed>" + Float.toString(speed) + "</speed>\r\n";
            body += "</position>\r\n";
            length = body.length();
            // costruzione dello URL di invocazione del WS
            url += URLEncoder.encode(targa, "UTF-8") + "?id=" + URLEncoder.encode(key, "UTF-8");
            URL server = new URL(url);
            HttpsURLConnection service = (HttpsURLConnection)server.openConnection();
            // impostazione degli header della richiesta HTTP di invocazione del WS
            service.setRequestProperty("Host", "https://car-ws.acme.com");
            service.setRequestProperty("Content-Type", "application/xml");
            service.setRequestProperty("Content-Length", Integer.toString(length));
            // impostazione del metodo HTTP di invocazione del WS (POST)
            service.setDoOutput(true);
            service.setRequestMethod("POST");
            // streaming del body XML
            OutputStreamWriter stream = new
                                        OutputStreamWriter(service.getOutputStream(), "UTF-8");
            BufferedWriter output = new BufferedWriter(stream);
            output.write(body);
            output.flush();
            output.close();
            // invocazione del WS
            service.connect();
            // codice di stato della risposta del WS
            status = service.getResponseCode();
        }
        catch (IOException exception) {}
    }

    public int getStatus() {
        return status;
    }
};
```

Volendo creare un oggetto istanza della classe *SendPosition* nel codice di un metodo di una *activity* è necessario che il costruttore sia invocato in un thread separato da quello principale della APP; a questo scopo è possibile incapsulare *SendPosition* in una classe Java che implementa un thread:

```
public class SendPositionThread extends Thread {
    volatile boolean done;
    volatile boolean error;
    String targa;
    String key;
    float latitude;
    float longitude;
    float speed;

    public SendPositionThread(String targa, String key,
                              float latitude, float longitude, float speed) {
         this.targa = targa;
         this.key = key;
         this.latitude = latitude;
         this.longitude = longitude;
         this.speed = speed;
         this.done = false;
         this.error = true;
    }

    public boolean getDone() {
         return done;
    }

    public boolean getError() {
         return error;
    }

    public void run() {
         SendPosition sendposition = new SendPosition(targa, key, latitude, longitude, speed);
         if (sendposition.getStatus() == 200)
            error = false;
         done = true;
    }
};
```

3)

Una rete Intranet è una rete aziendale che espone servizi accessibili alle applicazioni software ed al personale della stessa azienda utilizzando i protocolli normalmente utilizzati per la rete Internet, ad esempio: HTTP/HTTPS per rendere disponibili pagine web interattive e web-service, FTP per consentire l'archiviazione ed il recupero di file, SMTP/POP/IMAP per la gestione della posta elettronica, …

In **[2]** alle pagine 37 – 58 e in **[3]** alle pagine 1283 – 1313 sono trattati alcuni protocolli applicativi; in **[4]** alle pagine 24-31 è trattato in modo specifico il protocollo HTTP.

4)

Nella seguente tabella sono confrontati i vantaggi e gli svantaggi di una soluzione basata su infrastruttura interna (server fisici collocati nella DMZ della rete aziendale) e di una soluzione *cloud* (server virtuali resi disponibili da un data-center):

Infrastruttura interna ad ACME	***Cloud* in data-center esterno ad ACME**
Vantaggi:	**Vantaggi:** ⊙ nessun costo iniziale

	⊙ costo di mantenimento sempre adeguato alle necessità del servizio effettivamente offerto: le risorse del/i server virtuale/i (spazio di *storage*, capacità di elaborazione, banda di connettività Internet, …) sono adeguate continuamente e dinamicamente in funzione del numero di aziende servite e del numero di automezzi monitorati ⊙ esposizione di sicurezza del servizio erogato e dei dati memorizzati mitigata dal monitoraggio continuo e dalle misure adottate dal data-center ⊙ affidabilità in caso di guasti o incidenti garantita dal data-center
Svantaggi: ⊙ alto costo iniziale dovuto all'acquisto e alloggiamento dei server in locale sicuro e climatizzato, all'allestimento di un'infrastruttura di rete opportunamente ridondata e con alimentazione garantita per un adeguato periodo di tempo anche in caso di interruzione dell'erogazione dell'energia elettrica ⊙ alto costo di mantenimento dovuto alla presenza/reperibilità continuativa di personale tecnico specializzato per pronto intervento a fronte di problemi hardware e/o software e al noleggio di connettività Internet ridondata e con banda larga in *upstream* ⊙ necessità di prevedere risorse (spazio di *storage*, capacità di elaborazione, banda di connettività Internet, …) in quantità superiore alle reali necessità per poter far fronte all'espansione del numero di aziende servite e del numero di automezzi monitorati ⊙ elevata esposizione di sicurezza del servizio erogato e dei dati memorizzati ⊙ affidabilità in caso di guasti o incidenti comunque limitata	**Svantaggi:**

Riferimenti bibliografici

[1] F. Formichi e G. Meini, “Corso di informatica”, vol. 3, Zanichelli editore, 2013

[2] P. Ollari, “Corso di sistemi e reti”, vol. 3, Zanichelli editore, 2013

[3] A. Liberatore, M.L. Ferrario, G. Meini, F. Formichi, O. Bertazioli, G. Naldi e L. Marcheselli (a cura di), “Manuale Cremonese di informatica e telecomunicazioni”, Zanichelli editore, 2015

[4] G. Meini, F. Formichi, “Tecnologie e progettazione di sistemi informatici e di telecomunicazioni”, vol. 3, Zanichelli editore, 2014

