

# SAVE MY BIKE



  
SISTEMI DI SICUREZZA  
E INCENTIVAZIONE

## Documento di design della infrastruttura scalabile di back-end

O.O.2	Disegno SW (UI, Dashboard, etc..)	Azione	2.3
Partner Responsabile	GEOSOLUTIONS S.A.S.		
Autori	Lorenzo Pini, Simone Giannecchini		
Nome del file	Deliverable_2.3.1.pdf	Stato	<b>definitivo</b>

### Cronologia di approvazione del documento:

Data	Stato (Bozza/Revisione/Finale)	Autore/Revisore
28.07.2017	Bozza	Lorenzo Pini, Simone Gianneccchini
31.07.2017	Revisione/finale	Simone Gianneccchini

### Scopo del documento

Obiettivo di questo documento è fornire un primo design della infrastruttura scalabile di back-end a seguito degli studi e dei prototipi implementati ad inizio progetto per selezionare le scelte architettoniche e tecnologiche più adatte alla creazione di una serverless architecture scalabile e robusta che ci permetta di gestire l'ingestion e il processing delle tracce GPS registrate dall'App una volta in uso continuo dagli utenti, così come i calcoli e le notifiche relative ai premi.

### Destinatari del documento

- OP Leaders
- Partners
- Associates
- Stakeholders
- Decision Makers
- Altri \_\_\_\_\_

### Tipo di documento

- Private
- Non private
- Public

## INDICE

# Documento di design della infrastruttura scalabile di back-end

1	INTRODUZIONE .....	4
2	MODELLO DATI .....	4
2.1	ENTITÀ PRINCIPALI .....	4
2.2	RELAZIONI PRINCIPALI .....	5
2.3	SORGENTI INFORMAZIONI PRINCIPALI .....	6
3	MODELLO ASTRATTO DEL SISTEMA .....	6
4	MODELLO CONCRETO DEL SISTEMA .....	7
4.1	AUTENTICAZIONE E SINGLE-SIGN-ON CON KEYCLOAK .....	7
4.2	DBMS GEOSPAZIALE .....	8
4.3	UPLOAD E STORAGE DELLE TRACCE .....	8
4.4	BIGDATAENGINE: IL PROCESSAMENTO DELLE TRACCE .....	8
4.5	API DI ACCESSO AL MODELLO .....	8
5	CONCLUSIONI E PROSSIMI PASSI .....	9

## 1 INTRODUZIONE

Il progetto SaveMyBike in questa fase propone un primo design della infrastruttura scalabile di back-end; ciò a seguito degli studi e dei prototipi implementati ad inizio progetto per selezionare le scelte architettoniche e tecnologiche più adatte alla creazione di una serverless architecture scalabile e robusta tale da gestire sia l'ingestion e il processing delle tracce GPS registrate dall'App (una volta in uso continuo dagli utenti) che i calcoli e le notifiche relative ai premi.

## 2 MODELLO DATI

Prima di addentrarci nelle scelte relative ai componenti della piattaforma riteniamo sia opportuno soffermarsi a discutere quelle che sono i principali elementi del modello dati relativo al sistema di back-end a supporto del progetto SaveMyBike.

### 2.1 ENTITÀ PRINCIPALI

I concetti principali modellizzati dall'applicativo e dai servizi SMB e GoodGo sono i seguenti:

- **Utente** e relativi ruoli che ne definiscono i permessi e le autorizzazioni di accesso ai servizi. Rappresenta l'entità centrale attorno alla quale ruotano tutte le altre entità ed attività.
- **Bicicletta**: rappresenta le biciclette registrate dagli utenti. Ad ogni bicicletta è associato uno stato "in possesso" o "smarrita".
- **Rilevazione bicicletta**: rappresenta le informazioni relative alla rilevazione della posizione di una bicicletta tramite uno dei tag RFID associati
- **Tag**: tag RFID registrati ed associati ad una bicicletta
- **Sessione registrazione**: rappresenta una sessione di registrazione di una traccia dall'app mobile
- **Traccia**: Un insieme di segmenti registrati all'interno della stessa sessione di registrazione.
- **Segmento traccia**: Rappresenta un'insieme di waypoint consecutivi aventi tutto lo stesso tipo di mezzo assegnato (bici, auto, treno, ecc.). Ad ogni segmento sono associati valori degli indici di salute, d'impatto ecologico, ecc. previsti dalla piattaforma.
- **Waypoint**: rappresenta il singolo punto GPS registrato dall'app mobile durante una sessione di registrazione, e i valori dei sensori rilevati dal device mobile (accelerometro, giroscopio, altitudine, temperatura, ecc.)
- **Competizione**: rappresenta un tipo di competizione attiva in una data finestra temporale. Ogni competizione è associata ad un criterio di premialità prefissato. E' prevista la possibilità di partizionare la competizione per fasce di età.



- **Premio**: un premio, offerto da uno sponsor, che può essere connesso ad una o più competizioni.
- **Ledger**: rappresenta il “podio” degli utenti vincitori delle competizioni, e ne mantiene la storia.

Il modello sarà tradotto e normalizzato in un modello ER, per la sua implementazione su DB relazionale. Ulteriori entità accessorie andranno a completare i requisiti non funzionali della piattaforma.

## 2.2 RELAZIONI PRINCIPALI

Le seguenti relazioni costituiscono il fondamento del modello informativo della piattaforma, suddiviso in base ai principali servizi forniti da SaveMyBike (figura 2.1).

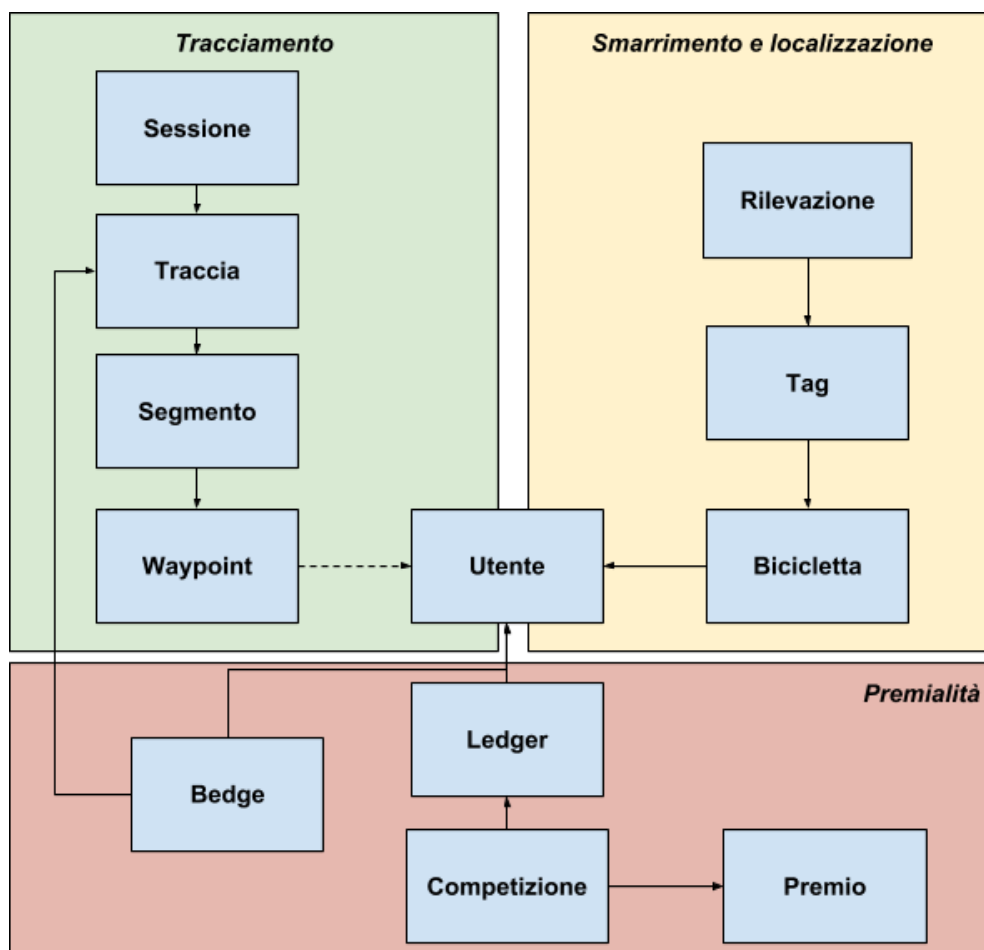


Figura 2.1: modello informativo della piattaforma

L'utente rappresenta l'elemento centrale che raccorda tutti le componenti della piattaforma.

Tuttavia, il sistema di gestione delle tracce, pur essendo necessariamente in relazione diretta con l'utente, sarà strutturato in modo da poter fornire i dati con i riferimenti all'utente solo all'utente stesso e a specifici utenti analisti con privilegi elevati.



Tutti i servizi accessori di analisi e visualizzazione dei dati forniranno soltanto viste aggregate e anonimizzate dei dati, senza possibilità di ottenere dati e riferimenti dei singoli utenti.

### 2.3 SORGENTI INFORMAZIONI PRINCIPALI

Le sorgenti principali della piattaforma sono rappresentate dai:

- sensori RFID mobili o fissi (antenne, gates), che forniscono le rilevazioni delle biciclette (registrate solo se in stato “smarrita”)
- device mobile degli utenti, quando gli utenti decidono di avviare una sessione di registrazione dei propri spostamenti (indicando il mezzo di trasporto attuale) e in seguito inviano la traccia al server.

Sui device mobile non è previsto alcun tipo di registrazione “passiva”, ovvero tramite servizi in background, non esplicitamente avviati dall’utente.

## 3 MODELLO ASTRATTO DEL SISTEMA

Il modello concettuale che descriva la nostra idea di piattaforma è mostrato in figura 3.1:

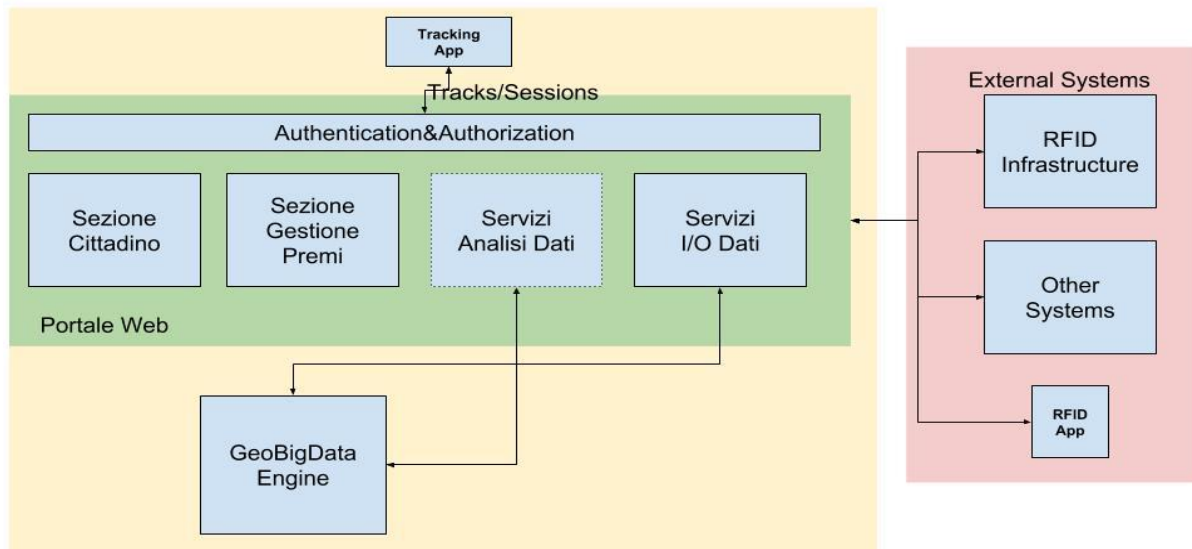


Figura 3.1: Modello concettuale della piattaforma da implementare

Abbiamo evidenziato innanzitutto una prima distinzione tra i vari sistemi con cui dovremo andare ad interagire ed a scambiare dati:

la app di tracking, che sarà responsabile per la registrazione e l’invio delle tracce degli spostamenti degli utenti  
gli applicativi esterni responsabili per le eventuali misurazioni spot come gate RFID o RFID reader manuali

Il sistema dovrà consentire l'invio massivo e continuo delle tracce GPS registrata dall'app quando in funzione nelle mani dei vari utenti finali ma anche permettere ad utenti applicativi selezionati e riconoscibili di inviare informazioni di più alto livello relative a furti di bici, ritrovamenti ed altre informazioni che sono appannaggio dei vari sistemi RFID. Per questo secondo gruppo di utenti/applicazioni si predisporrà una API REST protetta da autenticazione ed autorizzazione che permetta di interagire con il modello del sistema in modo controllato, per quanto riguarda la prima classe di utenti si farà invece ricorso a degli endpoint specifici che permettano di uploadare e processare le tracce in modo asincrono per rendere il sistema di ingestione robusto e scalabile.

I calcoli relativi ai premi ed alle gare saranno implementati da un apposito motore scalabile bigdata che lavorerà in modo asincrono rispetto agli upload di dati e provvederà ad aggiornare il modello dati. Il motore Big Data si appoggerà ad un database geospaziale per la registrazione dei dati ed il loro processamento geospaziale.

#### 4 MODELLO CONCRETO DEL SISTEMA

In Figura 4.1 è mostrata la traduzione in un modello concreto basato principalmente sulle tecnologie di AWS del modello astratto di cui sopra. Questo disegno è il risultato dei prototipi sviluppati durante la parte iniziale di progetto che ci hanno portato a validare queste scelte tecniche ed a creare le conoscenze necessarie ai successivi step implementativi.

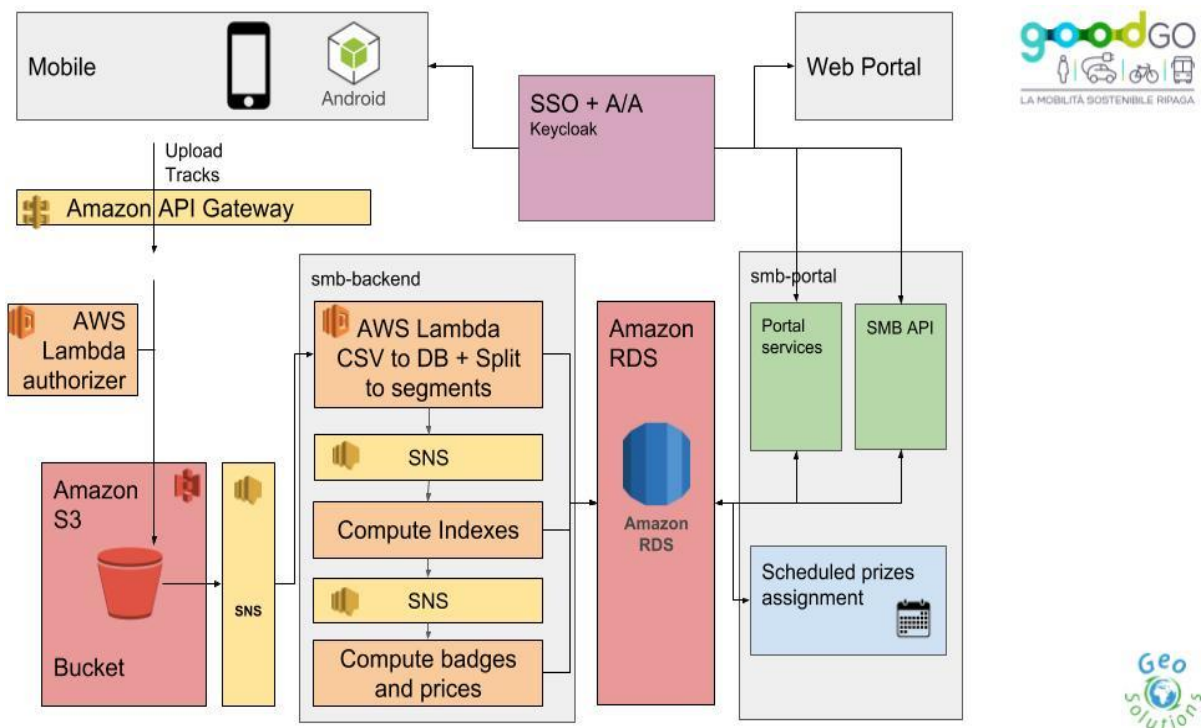


Figura 4.1: modello concreto basato sulle tecnologie serverless di AWS

##### 4.1 AUTENTICAZIONE E SINGLE-SIGN-ON CON KEYCLOAK

Per quanto riguarda i servizi di Autenticazione e di Single Sign On si è deciso di investire nell'utilizzo del component Open Source denominato Keycloak (<https://www.keycloak.org/>), vista la sua flessibilità e diffusione; nel nostro prototipo iniziale si è andati a lavorare con il servizio proprio di Amazon denominato Cognito (<https://aws.amazon.com/cognito/>) il quale si integra in maniera trasparente con la piattaforma, tuttavia a seguito di ulteriori test di integrazione con alcuni componenti che si è pensato possano essere utili in evoluzioni



future della piattaforma si è deciso di ripiegare su Keycloak vista la sua maggiore flessibilità e la possibilità di riuso anche in ambito diverso da AWS (Cognito è invece disponibile solo sulla piattaforma AWS).

## 4.2 DBMS GEOSPAZIALE

Come si evince in figura 4.1, la infrastruttura di back-end è divisa in due parti principali dal database geospaziale, la parte di ingestione e processamento da quella di disseminazione verso la App, il portale web e le applicazioni esterne (e.g. gestione RFID) basata su API REST autenticate.

La scelta del DBMS geospaziale caduta giocoforza su PostgreSQL, con le sue estensioni geospaziali PostGIS, per supportare in modo ottimale la ingestione ed il processing per il calcolo delle metriche di risparmio di CO2 relativi ai singoli utenti ma anche in vista di possibili estensioni future relative alla visualizzazione dei dati. In particolare, si è deciso di fare uso del servizio managed RDS messo a disposizione direttamente dalla piattaforma dei servizi cloud di Amazon in modo da semplificare l'installazione e la gestione del sistema.

## 4.3 UPLOAD E STORAGE DELLE TRACCE

Per quanto riguarda le funzionalità di upload delle tracce si è deciso per questioni di scalabilità del sistema di far uso delle funzionalità di API Gateway (<https://aws.amazon.com/api-gateway/>) messe a disposizione dalla piattaforma Amazon Web Service per creare una API REST managed e scalabile da usare tramite le App per fare upload diretto delle tracce in continua in formato binario compresso. Il codice esposto dall'API gateway andrà a salvare direttamente nello storage scalable S3 (Simple Storage Service - <https://aws.amazon.com/s3/>) i file binari delle tracce in modo scalabile, sicuro ed efficiente.

Il processamento delle tracce ingerite viene attivato in maniera asincrona rispetto all'upload tramite notifiche lanciate da trigger di S3 sul sistema Simple Notification Service (SNS - <https://aws.amazon.com/sns/>) di Amazon. In questo modo il sistema di upload e storage delle tracce è scalabile, serverless, economico e semplice da gestire in quanto quasi completamente managed dalla piattaforma AWS.

## 4.4 BIGDATAENGINE: IL PROCESSAMENTO DELLE TRACCE

Il motore di processamento delle tracce agirà in modo asincrono rispetto all'upload delle stesse in modo da assicurarne robustezza e scalabilità; si userà una infrastruttura serverless tramite delle lambda functions scritte in Python (come da prototipi) che andranno a calcolare a seguito di specifici eventi notificati dal sistema SNS tutti i parametri che guidano il sistema delle metriche, dei badge e dei premi. L'output finale sarà salvato in maniera sicura nel DBMS di Amazon di cui si è detto in precedenza.

## 4.5 API DI ACCESSO AL MODELLO

Una parte del sistema sarà dedicata a fornire delle API REST che ci permettano di esporre il modello interno dei dati sia al portale di progetto, che alla App che agli applicativi esterni. Per queste API si è deciso di non far ricorso all'API Gateway di Amazon visto la complessità del modello dati che dobbiamo andare a gestire ma si andrà a sviluppare le API REST usando framework come Flask o Django REST Framework direttamente sul database PostgreSQL di sistema, interagendo con il modello dati di cui sopra.



## 5 CONCLUSIONI E PROSSIMI PASSI

Nei capitoli precedenti si sono presentate le scelte di design effettuare in questa prima fase di progetto per quanto riguarda i componenti e le tecnologie da adottare a seguito dei numerosi test e prototipi effettuati.

Ovviamente ci aspettiamo degli scostamenti da questo documento perché ci stiamo muovendo su tecnologie per noi nuove con lo scopo di creare una applicazione per quanto possibile cloud native e scalabile; tuttavia pensiamo che in linea di massima le scelte tecnologiche fatte siano coerenti e consistenti, rimane tuttavia da metterle alla prova sul campo dopo aver implementato la business logic di progetto per la gestione dei calcoli e dei premi.