

Università degli Studi di Modena e Reggio Emilia

Dipartimento di Scienze Fisiche, Informatiche e Matematiche

Corso di laurea in Informatica

ERS

Anno accademico 2018/2019

1 Presentazione della situazione esistente

1.1 Engim srl

Engim è una piccola azienda informatica che ha due prodotti principali: servizioGPS e TwiceTouch. ServizioGPS è un servizio di noleggio di tracker gps ed è il prodotto di punta. La maggior parte dei clienti sono comuni che usano i nostri dispositivi per tracciare il lavoro delle macchine pulisci neve e spargi sale. I tracker sono sia dispositivi fisici da collegare al veicolo che un app per android. Twicetouch è un servizio di noleggio di dispositivi di sicurezza individuale. Il prodotto serve per i lavoratori in solitario, nel evenienza di un qualsiasi incidente viene inviata una segnalazione a dei numeri preposti. Questo segnalazione avviene nel caso in cui venga avvertito un grosso urto, oppure il lavoratore è fermo per un periodo prolungato, presumibilmente svenuto. Simile a servizioGPS c'è un prodotto fisico, il badge di TwiceTouch e un app per telefono.

1.2 Infrastruttura

Le tecnologie usate per servizioGPS sono le seguenti:

- Ruby on rails full stack
- Mariadb e Redis come database
- Python come back end di supporto
- Java per i prodotti app

Il servizio è inoltre diviso su molteplici server per evitare sovraccarico nelle giornate invernali, quando arrivato migliaia di punti contemporaneamente. Recente-

mente è stato introdotto un server in AWS, aprendo la possibilità di estrarre logica comune a tutti i server tramite le lambda e quindi rimuovere il peso dai singoli server.

1.3 Microservizi

In alcuni casi non è necessario avere un servizio costantemente acceso che gira, nello specifico, se devo fare alcune operazioni periodiche o in maniera discontinua allora non è utile avere un servizio o un api che è sempre in ascolto per un input. Ecco che entrano in gioco i Microservizi.

A differenza dei servizi, un microservizio non ha un suo server, non ha un suo ambiente e non dipende da nessun'altra tecnologia. Il Microservizio si crea l'ambiente ogni volta che viene richiamato (cold start), e rimane attivo per un lasso di tempo, svolge una specifica funzione, definita da un API precisa, e ritorna un messaggio HTML. Diventa molto vantaggiosa tale operazione soprattutto nel ambito web dove è molto naturale usare delle api per fare specifiche operazioni.

1.4 Sistema automatico di generazione di report

Le informazioni di ogni cliente vengono archiviate anno dopo anno. Nel caso in cui il cliente abbia necessità o interesse a salvare i dati del loro lavoro possono stampare o scaricare una serie di informazioni dal sito. La più importante stampa sono i punti che arrivano e le loro specifiche. Al momento questo è gestito internamente tramite una gemma di ruby. Il sistema attuale crea un file HTML, che viene poi convertito in pdf. Questo crea una serie di gravi problemi:

- Fare il parsing simili è estremamente costoso in maniera computazionale

- Per quanto è facile scrivere HTML il controllo sul prodotto finale è basso
- La conversione è molto lenta

Questa operazione inoltre si presta molto bene ai microservizi. È un'operazione ripetitiva, ben definita che ritorna sempre un file pdf. Quindi sarebbe possibile evitare calcoli extra di parsing e evitare di sovraccaricare il server se venisse esportata come microservizio. Inoltre una volta creata la logica per una singola stampa è possibile allargare la lambda per la stampa di un qualsiasi altro servizio.

2 Requisiti del progetto

2.1 Robe