



# Progettazione e sviluppo di una piattaforma di localizzazione dei testi per app e webapp

Stefano Dindo  
CEO  
[s.dindo@zero12.it](mailto:s.dindo@zero12.it)

Michele Massaro  
Project Manager  
[m.massaro@zero12.it](mailto:m.massaro@zero12.it)

# Chi siamo

“ Progettiamo e sviluppiamo soluzioni software innovative basate su tecnologia Cloud Amazon Web Services accompagnando i nostri clienti in un percorso di trasformazione digitale attraverso l'uso di metodologie agili. ”

# IL NOSTRO APPROCCIO

## Discover



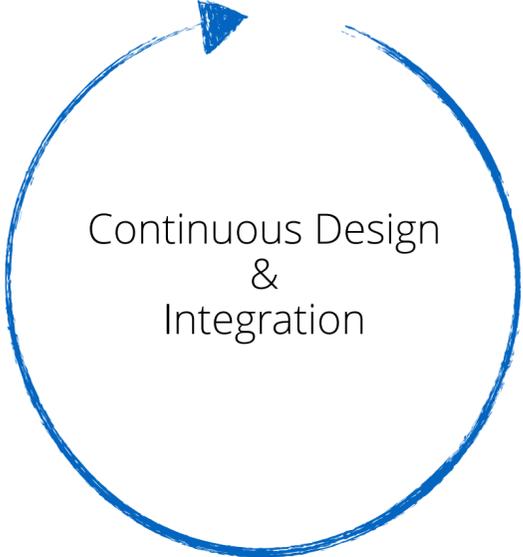
Customer & zero12  
collaboration meeting  
( CanvUX )



## Experiment



Idea



## Delivery



Users

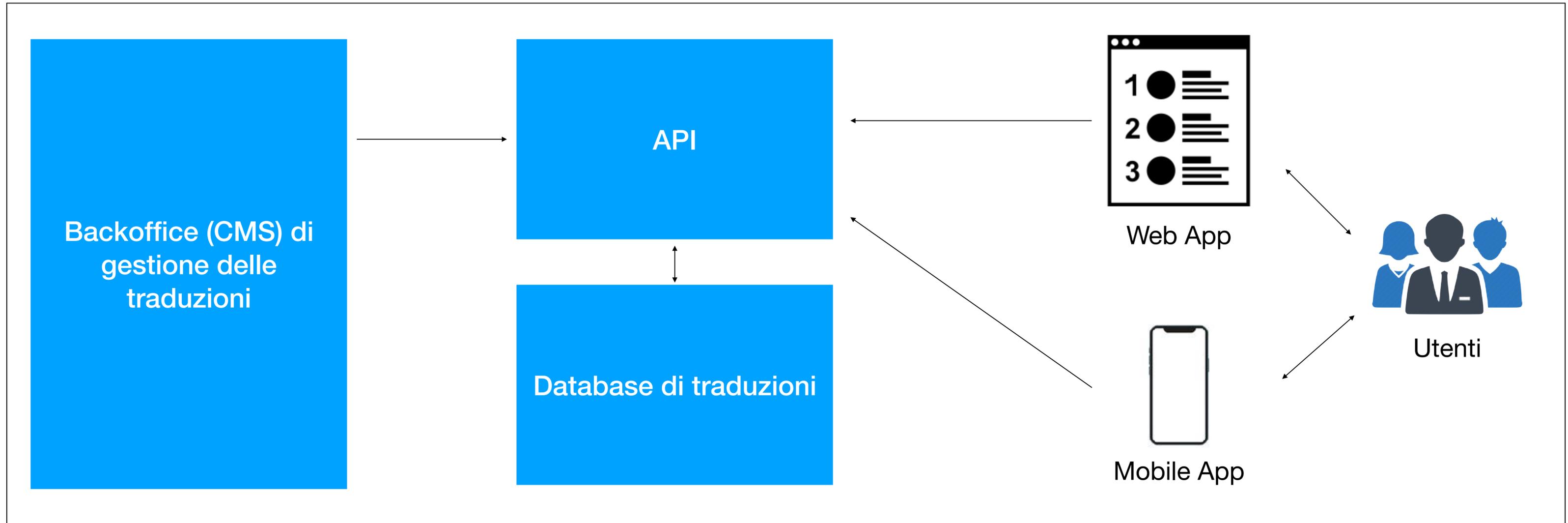


**CAPITOLATO**

# Problema

- **Molti applicativi moderni devono essere multi-lingua in modo da essere disponibili in un mercato internazionale**
- **Oltre all'implementazione tecnica, è necessario capire come gestire le traduzioni**
- **Come arrivano le traduzioni dall'agenzia che le crea al prodotto finale?**
- **Come si gestiscono gli update alle traduzioni?**

# Cosa vogliamo fare?



Multi-Tenant

Per Backoffice si intende un'interfaccia web di amministrazione accessibile solo al personale di zero12 o agenzie di traduzioni

Per Multi-Tenant si riferisce ad una architettura software in grado di erogare servizi per più organizzazioni mantenendo un isolamento logico dei dati tra le diverse organizzazioni

# Obiettivo

- Creare una piattaforma che permetta di gestire i testi delle localizzazioni di mobile app e webapp
- La piattaforma deve avere supporto multi-tenant
- Ci saranno due tipi di attori da considerare nello sviluppo che sono:
  - Admin user: sono gli che gestiranno le organizzazioni (temano) abilitate ad accedere alla piattaforma
  - Content user: utenti operativi che avranno il compito di creare contenuti e traduzioni
- Le traduzioni dovranno prevedere processi di approvazione
- Le traduzioni potranno avere un concetto di versionamento
- Le traduzioni saranno gestite tramite una chiave univoca
- Le traduzioni potranno essere suddivise in gruppi

# Tecnologie

Per questo progetto raccomandiamo di utilizzare la tecnologia di **Amazon Web Services** [1] ed in particolare i seguenti servizi:

- **AWS fargate** [2]: servizio serverless per gestione a container
- **AWS Aurora Serverless** [3]: servizio serverless di database SQL managed

Linguaggi di programmazione:

- **NodeJS** [4]: ideale per lo sviluppo di API Restful JSON a supporto dell'applicativo
- **Typescript** [5]: ideale per lo sviluppo di una libreria frontend
- **Swift** [6]: ideale per lo sviluppo di una libreria iOS/macOS
- **Kotlin** [7]: ideale per lo sviluppo di una libreria per ambiente Android

# Vincoli generali

Il team di progetto dovrà sviluppare una piattaforma che permetta di gestire le traduzioni per applicazioni mobile e webapp. Il risultato dovrà comprendere:

- Un'analisi di quali tecnologie cloud siano più adeguate per ottenere lo scopo, con una valutazione su quali permettano in modo migliore di gestire i carichi attesi.
- Un backoffice di amministrazione che permetta agli utenti admin di creare i tenant e agli utenti normali di creare le traduzioni gestendo i relativi processi di revisione e approvazione
- API a supporto di backend e client finali.
- Una libreria client che recupera le traduzioni in modo da usarle nella webapp o app. Il team può scegliere se creare una libreria per frontend web in Javascript/Typescript, per app iOS/MacOS in Swift o Android

Fornendo una giustificazione su ogni scelta condotta per lo sviluppo delle diverse componenti.

Alcuni vincoli del progetto:

- API sviluppate in **NodeJS**
- Architettura basata a **micro-servizi**. Tale modalità prevede di suddividere il progetto in tante funzioni di base. Ciascuna funzione, denominata servizio, può essere compilata e implementata in modo indipendente. Pertanto, i singoli servizi possono funzionare, o meno, senza compromettere gli altri.

**Sviluppo di una Web App** per il backoffice e una libreria JS/TS o Swift.

**Sviluppo di una libreria Mobile per iOS o Android**

**zero12 fornirà attività di formazione sulle principali tecnologie AWS e Mobile.**

# Materiali da condividere

Prima dell'inizio dell'attività di implementazione, il team dovrà condividere con zero12 i seguenti materiali:

- Diagrammi UML relativi agli Use Cases di progetto
- Schema Design relativo alla base dati
- Documentazione dettagliata di tutte le API in formato swagger
- Piano di test di unità

A valle del collaudo del progetto ("post live"), invece, il team dovrà condividere con zero12:

- Bug Reporting
- Codice prodotto in formato sorgente utilizzando sistemi di versionamento del codice come GIT o repository online ( github bitbucket etc)

*Gli studenti potranno esporre l'esperienza nel proprio CV e il codice sviluppato potrà essere usato da ogni singolo studente e dal tema di zero12.*

# Supporto zero12

Al fine di facilitare lo sviluppo del progetto il team di zero12 metterà a disposizione degli studenti interessati:

- Corso di formazione per lo sviluppo cloud native
- Corso di formazione per lo sviluppo di API Restful JSON secondo standard aziendali
- Canale diretto di comunicazione con il product owner identificato da zero12 per seguire il progetto . Il canale diretto di comunicazione verrà creato attraverso la definizione di un canale Slack dedicato al gruppo di lavoro
- \$ 500,00 di crediti per l'uso della piattaforma Amazon Web services dove elaborare il progetto
- Definizione di sessioni di Q&A durante la fase di sviluppo



**I corsi di formazione si potranno tenere da remoto o presso la nostra sede di Padova**

# Risorse informative

[1] **Amazon Web Services:** [aws.amazon.com](https://aws.amazon.com)

[2] **AWS Fargate:** <https://aws.amazon.com/it/fargate/>

[3] **Aws Aurora Serverless:** <https://aws.amazon.com/it/rds/aurora/serverless/>

[4] **Node.js:** <https://nodejs.org/it/>

[5] **TypeScript:** <https://www.typescriptlang.org/>

[6] **Swift:** <https://www.apple.com/it/swift/>

[7] **Kotlin:** <https://kotlinlang.org>



