

# Team: Canaries

## Challenge: Smoke Voice

Technische Informationen

### Komplette Github source codes

Embedded Code der auf unserem Canary läuft

[https://github.com/lumagician/smoke\\_canary\\_fw](https://github.com/lumagician/smoke_canary_fw)

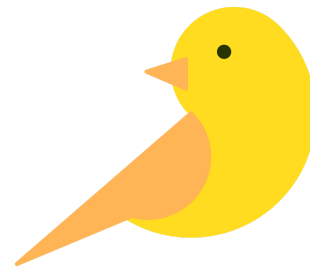
Der relevante Embedded Code ist in /src/main.cpp.

App Frontend, Backend Code

<https://github.com/AriJordan/canary>

Der Frontend Code ist unter /lib.

Der relevante Backend Code ist in /functions/src/index.ts.



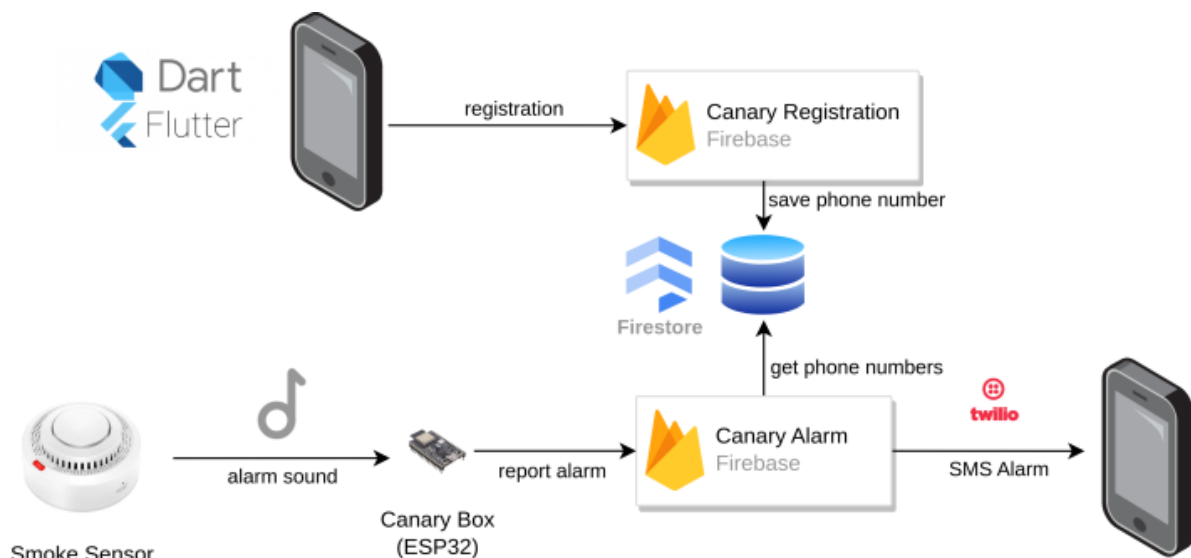
Canary

### Ausgangslage

Wir haben uns entschieden ein komplett funktionierendes Produkt zu bauen das die Challenge löst. Wir haben dafür am Anfang zusammen besprochen, geplant, abgeschätzt was realistisch ist, und dann entschieden was wir alles für unser Produkt brauchen. Wir haben uns für diese Teile entschieden, die wir auch alle umgesetzt haben:

- Unser "GVB Canary": ein ESP32 mit Mikrofon in einem 3D gedruckten Gehäuse, der Rauchalarme entdeckt
- Unsere App <https://gvb-canary.web.app> in Flutter
- Unser Backend das auf einem Firebase Server läuft

### Technischer Aufbau



Wenn unser Canary mit unserem Signalverarbeitungsalgorithmus einen Rauchalarm entdeckt schickt es ein HTTP Signal an unser Backend. Das Backend schickt mithilfe von Twilio dann an alle registrierte Nutzer des Canary's eine SMS. Nutzer können über unsere App <https://gvb-canary.web.app> mit ihrem Canary Gerät und ihrer Telefonnummer registrieren.

## Hardware

Wir haben uns entschieden unsere kosteneffiziente ESP32 mit Mikrofon als Hardware zu verwenden, damit wir wirklich ein gutes bezahlbares Produkt anbieten könnten.

## Frontend

Wir haben uns entschieden für unsere App Flutter zu verwenden weil wir damit eine Webapplikation bauen können, und mit kleinen Modifikationen auch eine Android und iOS App auf den Play Store beziehungsweise App Store zur Verfügung stellen könnten. Die App wir für die Registration des Gerätes und Telefonnummer des Nutzers angeboten.

## Backend

Für das Backend haben wir uns für Firebase entschieden, weil es einfach und sehr günstig ist, es schnell es auf einem Google Firebase Server zu hosten. Es benutzt eine Twilio Extension, womit wir das Schicken der SMS implementieren konnten.

## Implementation spezielles/cool

- Unsere Canary Hardware kostet weniger als 10 CHF.
- Unser Canary verbraucht extrem wenig Strom der weniger als 1 CHF pro Jahr kostet.
- Daten wie Audioaufnahmen werden lokal verarbeitet. Wir schicken nur die Gerätenummer und Telefonnummer auf unsere Server um die SMS zu schicken.
- Unser Produkt funktioniert weltweit, solange man einen Rauchmelder und unseren Canary hat.
- Pro SMS die durch unser Backend verschickt wird, müssen wir weniger als 0.1 CHF zahlen.
- Alles Design inklusive Logo und Canary Gehäuse haben wir selber erstellt.

## Abgrenzung / Offene Punkte

Wir haben uns auf den von der GVB gegebenen Rauchmelder fokussiert. Für andere Rauchmelder müssten wir wahrscheinlich den Signalverarbeitungsalgorithmus auf unserem Canary verbessern.