

Mit Oracle und Java für die Feuerwehr

Guy Saldanha
Rescue Call UG
Boitin-Resdorf

Schlüsselworte

Oracle, Java, Cloud, Mobile Services, Feuerwehr

Einleitung

Viele Freiwillige Feuerwehren haben insbesondere tagsüber das Problem, dass nach Alarmierung zunächst nicht klar ist, wie viele Einsatzkräfte tatsächlich zum Einsatz kommen (können).

Die Lösung - schnell und einfach Rückmeldungen sammeln und den Einsatzleitern bei der Einsatzplanung helfen - jede Minute zählt!

Sekundär-Alarmierung für Freiwillige Feuerwehren als Cloud-Service mit Oracle

Rescue Call (Englisch für Notruf) ist ein Dienst für die Freiwillige Feuerwehren in Deutschland.

Mit dem mobilen Dienst wird eine Sekundäralarmierung an die Feuerwehrmannschaft gesendet. Die Mannschaft kann über ihre mobilen Geräten schnell und einfach Rückmeldung geben, ob sie zum Einsatz kommen können.

Die Führungskräfte sehen auf ihren Geräten die Zusammenfassung, Qualifikation und Position der Einsatzkräfte und können so schnell entscheiden, welche Fahrzeuge wie besetzt werden können.

Damit wird wertvolle Zeit gewonnen und die Rettungskräfte können schneller ausrücken – jede Minute zählt!

Funktionsweise

Anmeldung und Einrichten der Mannschaft

Der Team-Admin (in der Regel ein Mitglied der Wehrführung) legt ein Konto auf www.rescue-call.de an und lädt seine Mannschaft per Mail ein. Über die Admin-Konsole kann er seine Mannschaft verwalten, z.B. Mitglieder hinzufügen und ändern, Rollen zuordnen, Smartphones der Mitglieder verwalten usw..

Es gibt drei Rollen von Benutzern:

- Team-Admin – Mitglieder und Smartphones verwalten, Rollen setzen, Lizenzen einsehen uvm.
- Team-Leiter – Alarmierungen auslösen, Alarm-Rückmeldungen und Positionen auf der Karte sehen u.a.
- Team-Mitglied – auf die Alarmierung reagieren, Profilbild setzen u.a.

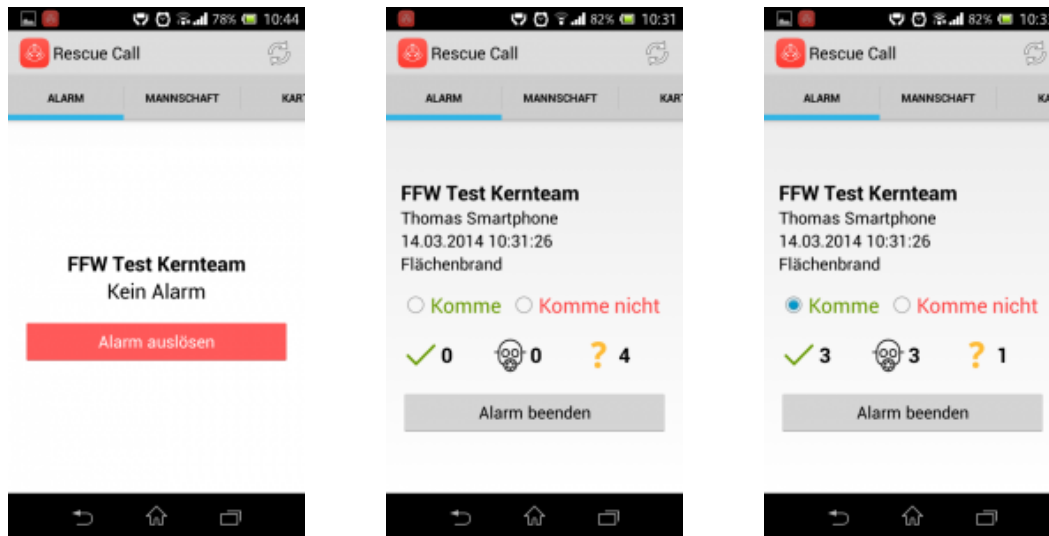
Die Rollen wurden absichtlich unabhängig von Feuerwehr-Rängen und Ausbildungsgraden ausgewählt, damit jede Feuerwehr die Rollen entsprechend ihres Alltags einrichten kann. So kann es z.B. bei kleineren Wehren zweckmässig sein, dass ein Truppführer als Team-Leiter eingetragen ist – bei größeren kann es Sinn machen, nur Zugführern diese Rolle zuzuordnen.

Auslösung eines Alarms

Der Pieper heult – Alarm! Der Team-Admin oder ein Team-Leiter öffnet die Rescue Call-App und tippt auf ‘Alarm auslösen’. Alle Mitglieder des Teams erhalten sofort eine Push-Nachricht auf ihr Smartphone, die folgende Informationen enthält:

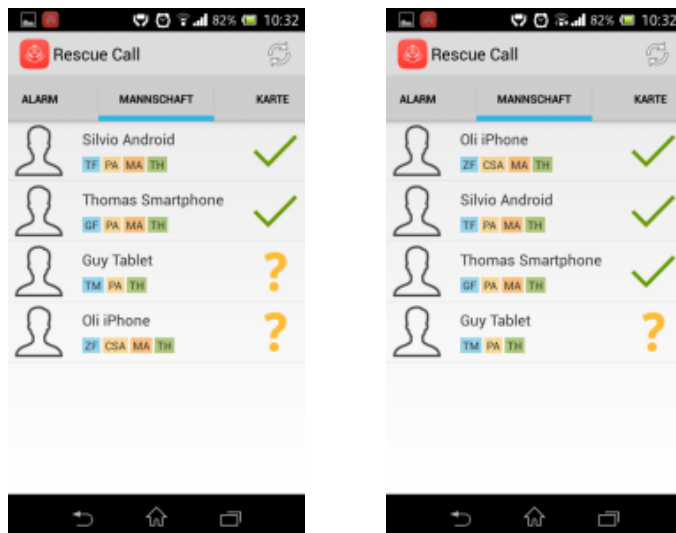
1. Name des Teams (in der Regel der Name der Feuerwehr)
2. Name des Mitgliedes, das den Alarm ausgelöst hat

3. Datum und Uhrzeit des Alarms
4. Einsatzstichwort (optional, kann bei der Auslösung vergeben werden).



Das Team-Mitglied wählt jetzt 'Komme' oder 'Komme nicht', die Anzeige aktualisiert sich dabei automatisch je nach eingegangenen Rückmeldungen. Der grüne Haken steht für die gesamten positiven Meldungen ('Komme'), die Gasmasken für die Anzahl der Atemschutzträger (mehr dazu in der Admin-Console) und das Fragezeichen für Mitglieder ohne Rückmeldung.

Auf dem Mannschfts-Tab sind Informationen über die Mitglieder des Teams abrufbar (Nutzername und -Bild, Rückmeldung, Ausbildungsstand):

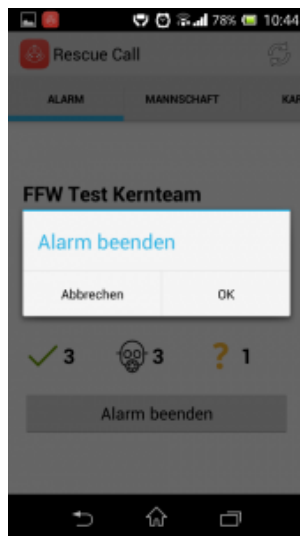
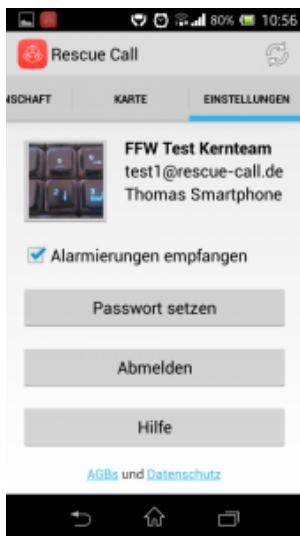


Auf dem Karten-Tab sind die Positionen der Mitglieder einsehbar, die sich mit 'Komme' zurückgemeldet haben. Alle anderen Mitglieder sind auf der Karte nicht sichtbar. (Die Positionsdaten

werden nicht gespeichert, nach 30 Minuten erfolgt ein automatisches Alarm(ierungs)-Ende, damit sind auch diese Positionen nicht länger einsehbar.)



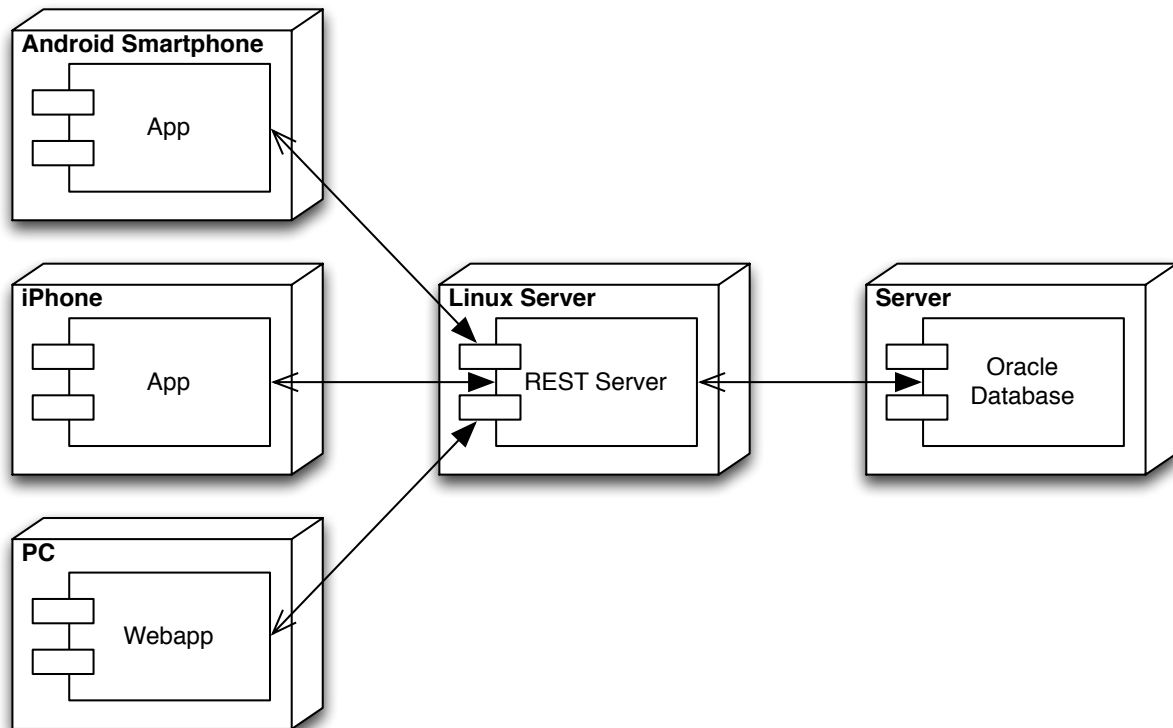
Unter Einstellungen lässt sich u.a. ein eigenes Benutzerbild anlegen und der Empfang der Alarmierungen deaktivieren – z.B. für Schichtarbeiter oder die Dauer der Urlaubsreise.



Die Alarmierung beendet sich automatisch nach 30 Minuten oder kann von einem Team-Admin oder -Leiter jederzeit vor Ablauf beendet werden.

Systemarchitektur

Für die technische Realisierung wurde eine typische Systemarchitektur für einen mobilen Dienst verwendet.



Apps auf Smartphones verbinden sich über HTTPS mit einem Server, der eine REST¹ Schnittstelle anbietet. Ressourcen (z.B. Userdaten) sind über URLs erreichbar. Anfragen werden mit den üblichen HTTP Befehlen gemacht (GET, POST, PUT, DELETE), Daten werden in JSON Format ausgetauscht.

Da der REST Server *Stateless* ist, lassen sich beliebig viele Instanzen hinter einem Load Balancer aufstellen.

Die REST Server speichern Daten in eine Datenbank. Hier ist auch der Engpass dieses Aufbaus, die Datenbank muss gut skalierbar sein. Zur Entlastung der Datenbank kann ein Server als Cache dienen, um z.B. Sessiondaten zu speichern.

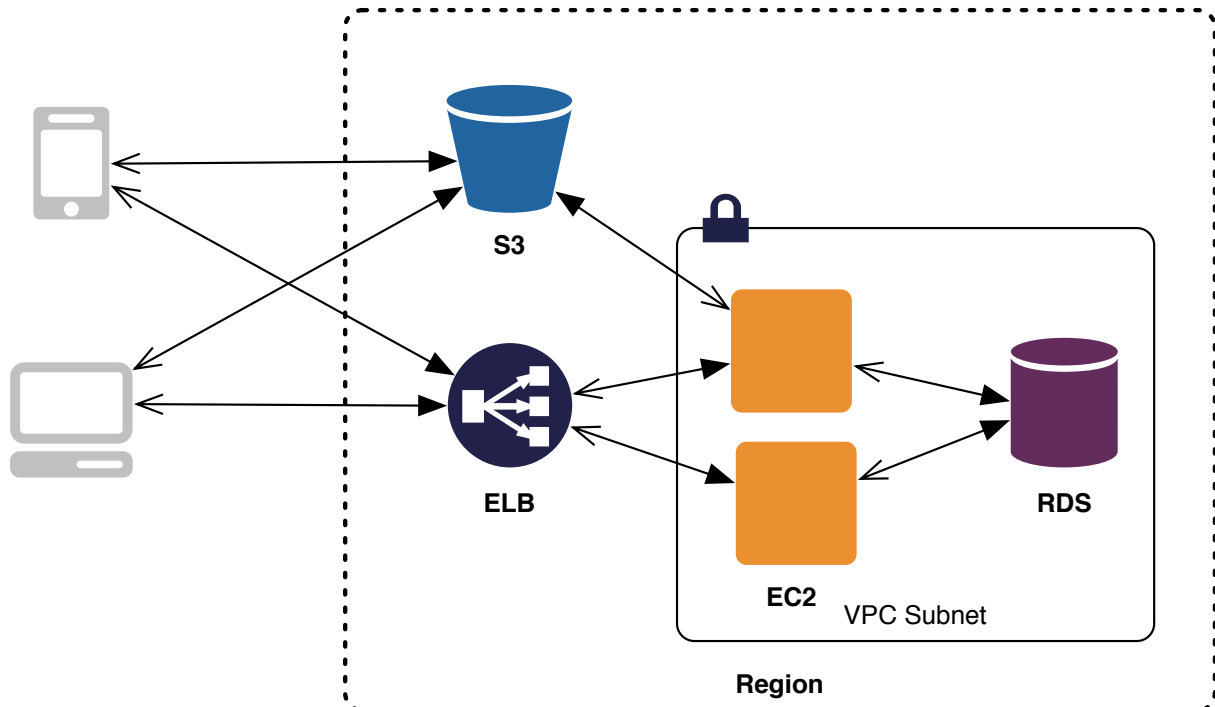
AWS als Infrastruktur

Zur Umsetzung der Architektur werden Amazon Web Services (AWS) verwendet. Hierbei kommen folgende Komponenten des IAAS zum Einsatz:

- EC2 Server. Auf diesen Server läuft Linux mit einem Java EE Container (s.u.).
- ELB Load Balancer. Anfragen der Clients gehen nicht direkt an die EC2 Server, sondern werden an einem Load Balancer geschickt, der diese an EC2 Server verteilt. Der Load Balancer can über CloudWatch die EC2 Instanzen monitoren und in Problemfällen (HTTP 500 oder Server nicht erreichbar) E-Mails verschicken.

¹ Representational State Transfer

- RDS Datenbank. Dieser Dienst bietet das einfache Verwalten von Datenbank Instanzen an. Bei Rescue Call wird eine Oracle 11g RDS Instanz verwendet.
- S3 Dateispeicher. Hier werden z.B. Profilbilder gespeichert.



Alle Dienste laufen in einer Region. Die EC2 und RDS Dienste sind vom Internet abgeschirmt.

Einsatz von Java EE

Java EE wird bei dem REST Server und der Admin Webapp verwendet.

REST Server

Der REST Server wurde in kurzer Zeit mit folgenden Java Technologien realisiert:

- JPA zur Persistenz der Modell Objekte
- JAXB zur Serialisierung der Modell Objekte (Umwandlung von und zu JSON für Requests)
- JAX-RS zur Definition der REST Schnittstellen

Die Entwicklung lief schnell und ohne Probleme.

Admin Webapp

Die Admin Webapp wurde mit folgenden Java Technologien realisiert:

- JSF als Web Framework
- Primefaces UI Komponenten
- Jersey Client zur Anbindung an den REST Server

JSF und Primefaces sind hilfreich, um schnell UI zusammen zu bauen, aber bei Umsetzung eigener Designs nicht einfach hand zu haben.

Glassfish

Der Glassfish Server wird seit Version 4.0 nicht mehr kommerziell von Oracle vertrieben. Updates des Servers werden nicht mehr so häufig veröffentlicht. Es war z.T. nötig, zwischen Releases den Server manuell zu patchen (Beispiel: GZIP Komprimierung).

Der JEE Server läuft nach dem "Start once run forever" Prinzip. Er läuft zwar stabil, startet aber sehr langsam. Aus diesem Grund konnte das Autoscaling von AWS nicht eingesetzt werden.

Einsatz von Oracle DB

Zur Persistenz der Daten wurde die Oracle Datenbank gewählt. Gründe hierfür sind viele Jahre Erfahrung mit Oracle Datenbanken und die gute Performance.

Es wird eine RDS Instanz mit Oracle 11g samt SE1 Lizenz bei AWS gemietet. Die Lizenzkosten können so auch von einem Startup getragen werden.

Die Datenbank wird automatisch gesichert, Snapshots können manuell angestoßen werden. Die Vergrößerung des verfügbaren RAMs und Anzahl der CPUs machen eine Skalierung einfach.

Kontaktadresse:

Guy Saldanha
Rescue Call UG (haftungsbeschränkt)
Dorfstr. 13
23923 Boitin-Resdorf
Deutschland

Telefon: +49 (0) 38821 68671
Fax: +49 (0) 38821 68674
E-Mail: info@rescue-call.de
Internet: www.rescue-call.de