

# Proiect Cloud Computing

Iulian Oleniuc, Elisabeta Dima

Mai 2023

## 1 Introducere

Am ales să creăm o aplicație minimalistă de mesagerie ce permite crearea de conversații și trimiterea de mesaje text între utilizatorii din cadrul aceleiași conversații.

Majoritatea aplicațiilor de chat fie sunt mult prea complexe și adaugă constant funcționalități neesențiale pentru mesagerie (Discord), fie prezintă bug-uri ce se fixează după foarte mult timp (Facebook Messenger), fie sunt suficient de economice din punct de vedere al resurselor folosite, dar nu prezintă o interfață prea plăcută de desktop/web (WhatsApp).

Așadar, am concluzionat că aplicația noastră de mesagerie poate fi o soluție viabilă care să depășească în popularitate soluțiile menționate mai devreme.

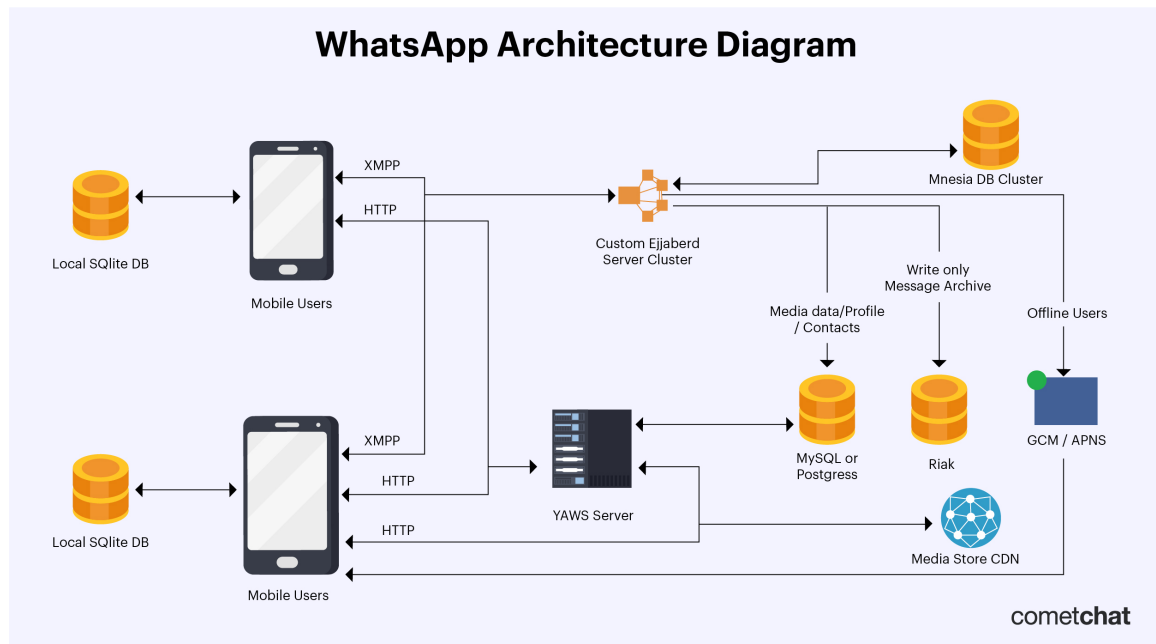
### 1.1 Soluții existente

În această secțiune vom analiza din perspectivă tehnică unele dintre cele mai populare soluții existente în momentul de față.

#### 1.1.1 WhatsApp

Sursă: [link](#).

- **Main Language:** Erlang
- **Operating System:** FreeBSD
- **Communication Protocol:** XMPP
- **XMPP Application Server:** Ejabberd
- **Erlang Virtual Machine:** BEAM
- **Erlang Database:** Mnesia
- **Multimedia Web Server:** YAWS



### 1.1.2 Discord

Sursă: [link](#).

- **Storage:** Scylla (în trecut, MongoDB și Cassandra)
- **Web Sockets:** Elixir
- **REST API:** Python & Flask
- **Server:** Go
- **Discord Store:** Rust
- **Discord RPC Library:** C++
- **Cloud Services:** Google Cloud Platform
- **Proxy:** Cloudflare
- **Search:** Algolia
- **Frontend:** Electron

## 2 Arhitectura aplicației

### 2.1 Tehnologii folosite

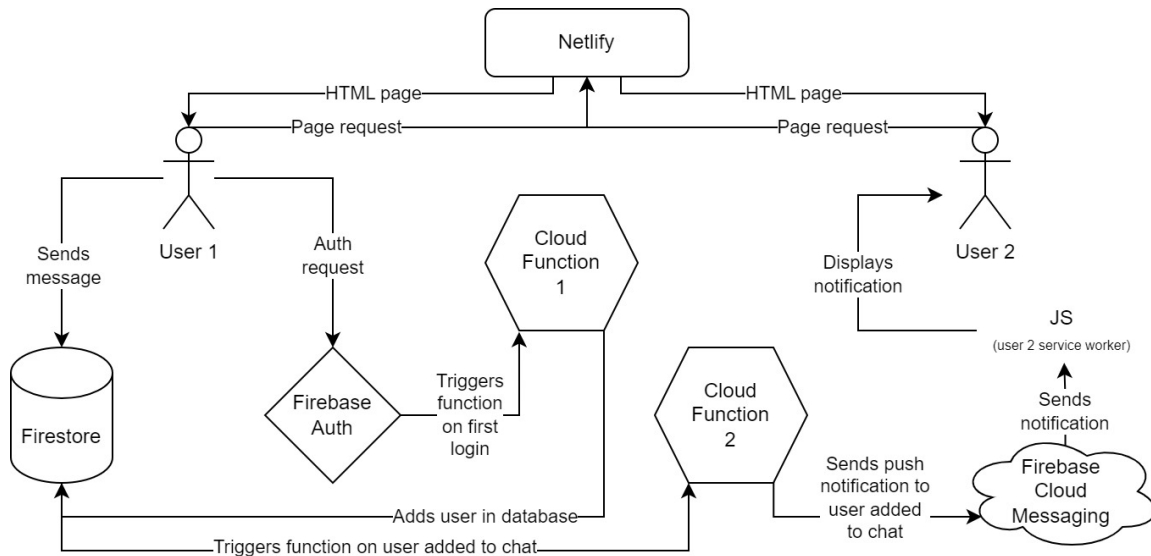
- **Frontend:** Vue 3
- **Backend:** NodeJS, Nuxt 3
- **Servicii:** Google Cloud Platform, Firebase, Stripe

### 2.2 Servicii cloud

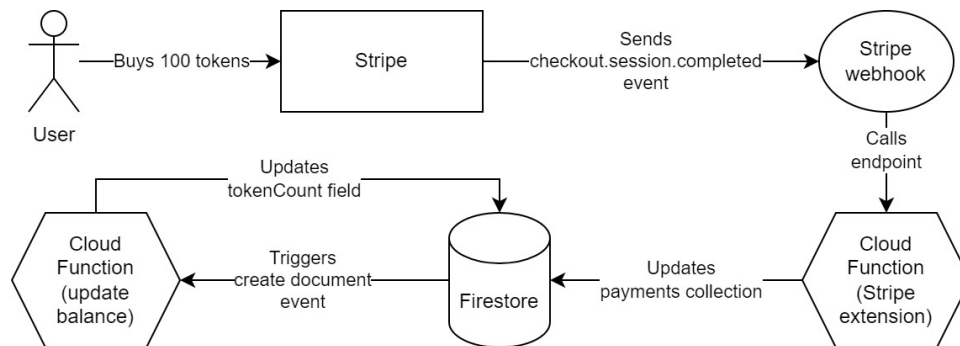
- **Firestore:** un serviciu de bază de date în cloud oferit de Google Cloud Platform care pune la dispoziție un sistem de stocare scalabil, flexibil și în timp real. Toate datele despre utilizatori și grupuri, precum și mesajele trimise, sunt stocate într-o bază de date.
- **Firebase Authentication:** un serviciu de autentificare în cloud de la Google Firebase care oferă o modalitate simplă și securizată de autentificare a utilizatorilor în aplicații mobile și web. În aplicația noastră, utilizatorii se pot autentifica cu contul de Google.
- **Firebase Cloud Messaging (FCM):** un serviciu de mesagerie push în cloud ce permite dezvoltatorilor să trimită notificări push, mesaje și date la aplicațiile mobile și web, chiar și atunci când acestea nu sunt active. Aplicația de mesagerie pe care am creat-o are opțiunea de a trimite notificări unui utilizator în momentul în care acesta este adăugat într-un grup nou.
- **Firebase Functions:** un serviciu care permite dezvoltatorilor să ruleze cod JavaScript (sau TypeScript) în cloud, fără a fi nevoie să gestioneze infrastructura serverelor. Acest serviciu poate fi folosit pentru a răspunde în timp real la evenimente din aplicație, ceea ce este util în cazul în care un utilizator este adăugat într-o conversație, primește un mesaj nou sau efectuează o plată prin Stripe.
- **Stripe:** un procesor de plăți online ce intermediază procesul de checkout dintre utilizatorul ce plătește folosind un card bancar și aplicația web. Integrarea Stripe cu Firebase – în speță cu Firebase Authentication și cu Firestore, unde sunt stocate colecții specifice precum **products** și **payments** – este facilitată de o extensie Firebase corespunzătoare.
- **Cloud Vision API:** un serviciu Google Cloud folosit, în cazul nostru, pentru etichetarea imaginilor trimise în chat. Mai precis, odată ce urcăm imaginea în Storage, Cloud Vision o analizează și salvează în Firestore 10 etichete cu obiecte/teme identificate în acea imagine.
- **Invertase Image Processing API:** o extensie Firebase care creează o funcție cloud ce primește o imagine neprocesată și returnează o versiune optimizată a sa, atât ca format (WebP) cât și din punct de vedere al calității sale (care este micșorată). De asemenea, după primul request la un URL anume, resursa respectivă este salvată în Storage (*cached*).
- **Netlify:** un serviciu pentru deploy-ul de aplicații web în cloud, care se ocupă în acest caz de servirea aplicației Nuxt către useri. În plus, deploy-ul se efectuează automat în urma fiecărui nou push în repo-ul de GitHub asociat aplicației, folosind *continuous integration*.

## 2.3 Diagrame use-case

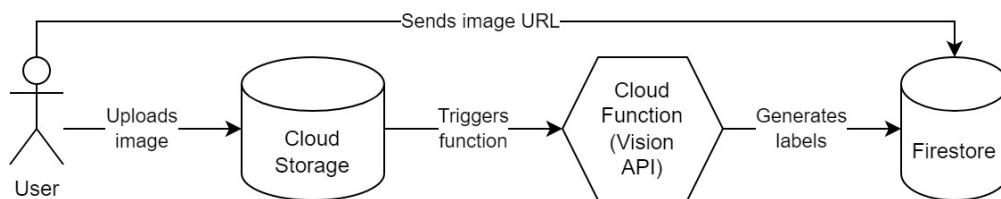
### 2.3.1 Crearea unui user nou și trimiterea de notificări pentru mesaje noi



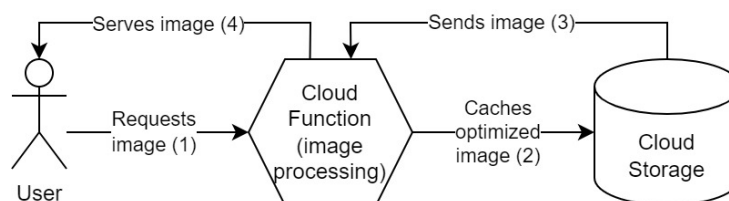
### 2.3.2 Efectuarea unei plăți prin Stripe



### 2.3.3 Generating image labels using AI



### 2.3.4 Optimizing chat images



### 3 Feature-uri

1. **LaTeX**: Nicio soluție populară de mesagerie nu permite utilizatorilor să trimită expresii  $\text{LaTeX}$  în mesajele lor. Însă aplicația noastră are acest feature! Expresiile  $\text{LaTeX}$  din cadrul unui mesaj trebuie încadrate între două semne  $\$$  (exemplu:  $\$E=mc^2\$$ ).

2. **API public**: Aplicația pune la dispoziție dezvoltatorilor un API minimal pentru crearea de boți de chat. Pentru utilizarea sa, utilizatorul trebuie să intre pe pagina `/bots` și să genereze o cheie API secretă (sau să regenereze una dacă cea anterioară a fost compromisă). Cheia API va fi trimisă în requesturile ulterioare, pentru autentificarea utilizatorului de către server.

Trimiterea unui mesaj de către un bot costă 1 token. Inițial, fiecare utilizator deține 10 tokeni, însă, ori de câte ori dorește, poate achiziționa între 1 și 10 pachete a câte 100 de tokeni fiecare, la prețul de 1\$.

Endpointul expus de către acest API este `PUT /api/message`, care primește în body trei câmpuri `apiKey`, `bot` și `message` și trimite mesajul `message` în chatul `bot` (id-ul botului este același cu id-ul chatului aferent lui).

3. **Imagini**: Pe lângă mesaje text, utilizatorii pot trimite și fotografii. Acestea sunt optimizate, pentru a putea fi încărcate repede chiar și pe dispozitive ce dispun de o conexiune slabă la internet. De asemenea, mesajele ce conțin fotografii sunt însoțite de o descriere formată din numele a 10 concepte (persoane, obiecte, culori, teme, însușiri etc.) observate de Cloud Vision API în acestea.

4. **Alte feature-uri**: Dacă utilizatorii doresc să modifice numele unui chat, acest lucru este posibil prin comanda `$ rename newChatName`, unde `newChatName` poate fi compus din mai multe cuvinte separate prin spații.

Un alt feature util este evidențierea adreselor URL printr-o culoare diferită și convertirea acestora în link-uri clickable.

### 4 Business canvas

