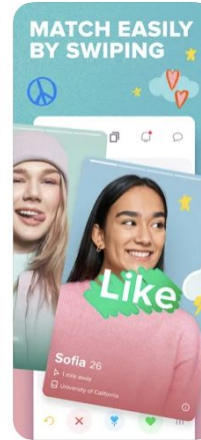
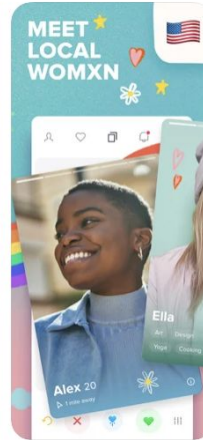
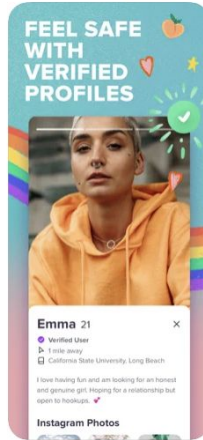
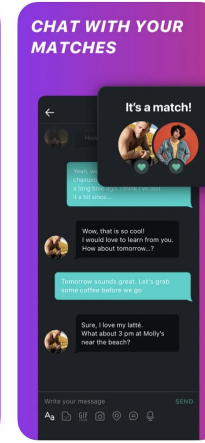
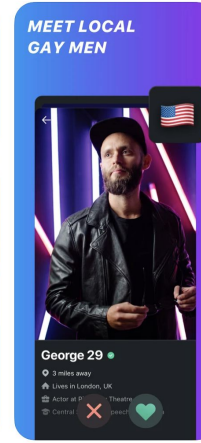
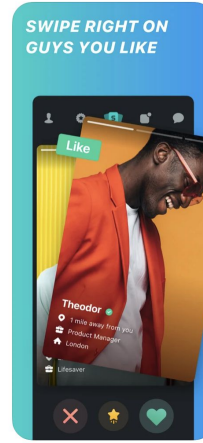
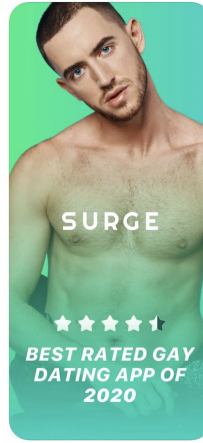


STRV DATING APPS |

Jozef Cipa, Backend Developer at STRV

STRV



TECHNICAL BACKGROUND

- iOS & Android
- Node.js & Typescript
- Runs on Heroku
- Data in AWS
 - RDS (Postgres)
 - S3
 - Cloudfront
 - **15TB** of data combined
- Firebase
 - Chat & real-time data
 - Push notifications



CHALLENGE #1 - MIGRATING DATABASE

- Originally Postgres database as a Heroku Addon
- **Scaling**
 - Cannot configure RAM, CPU, storage
 - Becomes **very expensive** 💰💰💰
- **Configuration**
 - Postgres VACUUM not configured correctly
 - Wrongly designed indexes
 - Results in 300GB of size

Hobby Dev	Free
Hobby Basic	\$9/mo
Standard 0	\$50/mo
Standard 2	\$200/mo
⋮	
Private 7	\$7000/mo
Shield 7	\$8400/mo
Premium 8	\$8500/mo
Private 8	\$10000/mo
Premium 9	\$11000/mo

and it goes higher 😬

CHALLENGE #1 - MIGRATING DATABASE

- Originally Postgres database as a Heroku Addon
- **Scaling**
 - Cannot configure RAM, CPU, storage
 - Becomes **very expensive** 💰💰💰
- **Configuration**
 - Postgres VACUUM not configured correctly
 - Wrongly designed indexes
 - Results in 300GB of size (~80GB after fixing the issues)

Hobby Dev	Free
Hobby Basic	\$9/mo
Standard 0	\$50/mo
Standard 2	\$200/mo
⋮	
Private 7	\$7000/mo
Shield 7	\$8400/mo
Premium 8	\$8500/mo
Private 8	\$10000/mo
Premium 9	\$11000/mo

and it goes higher 😬

CHALLENGE #1 - MIGRATING DATABASE



CHALLENGE #1 - MIGRATING DATABASE

Migrating to AWS RDS

- Better scaling options
- Reserved (prepaid) instance can save you up to 60% of cost
- Ideally via **failover mechanism** (add 2nd DB follower replica, promote as a new primary)
 - Not supported by Heroku
- Manually via *pg_dump* + *pg_restore* 💪
 - Results in 20 minutes application downtime 😞

CHALLENGE #2 - MIGRATING ALL USER FILES

- All data originally stored in Microsoft Azure
- Need to migrate it to AWS (legacy, pricing, “everything under one roof” reasons)
- Photos & chat files

Steps

1. Update the API to upload new files to AWS
2. Move all photos to AWS & update DB records
3. Move all chat media to AWS & update Firebase records





CHALLENGE #2 - MIGRATING ALL USER FILES

- Running on **Heroku**
 - ~700 user profiles / hour, est. completion time: **~8 months**
✗ *time*

CHALLENGE #2 - MIGRATING ALL USER FILES

- Running on **Heroku**
 - ~700 user profiles / hour, est. completion time: **~8 months**
✗ *time*
- Rewriting script to leverage **Node.js cluster** (with 3 parallel workers)
 - ~3 900 user profiles / hour, est. completion time: **~1.5 month**
✗ *resources, price*
⚠ *multiple workers orchestration, respawning crashed processes*

CHALLENGE #2 - MIGRATING ALL USER FILES

- Running on **Heroku**
 - ~700 user profiles / hour, est. completion time: **~8 months**
 *time*
- Rewriting script to leverage **Node.js cluster** (with 3 parallel workers)
 - ~3 900 user profiles / hour, est. completion time: **~1.5 month**
 *resources, price*
 *multiple workers orchestration, respawning crashed processes*
- Moving the script to **AWS EC2** (with 6 parallel workers)
 - ~24 000 user profiles / hour, est. completion time: **~7 days**
 *resources, price, time*

CHALLENGE #2 - MIGRATING ALL USER FILES

- Running on **Heroku**
 - ~700 user profiles / hour, est. completion time: **~8 months**
✗ *time*
- Rewriting script to leverage **Node.js cluster** (with 3 parallel workers)
 - ~3 900 user profiles / hour, est. completion time: **~1.5 month**
✗ *resources, price*
⚠ *multiple workers orchestration, respawning crashed processes*
- Moving the script to **AWS EC2** (with 6 parallel workers)
 - ~24 000 user profiles / hour, est. completion time: **~7 days**
✓ *resources, price, time*

 [surgeapp/parallel-worker](https://github.com/surgeapp/parallel-worker) for the rescue to all **⚠**

CHALLENGE #3 - DUPLICATE HASHES

- User registers => create a hash (5 hexadecimal characters)
- SQL procedure writing to tables and generating hash

Problem

- Some requests took 30s and got cut off by Heroku
- SQL queries stuck in Postgres (no timeouts configured)
- Out of hashes after 1M+ ($16^5 = 1\,048\,576$ unique combinations)

```
while(!foundHash) {  
    foundHash = generateHash()  
}
```

CHALLENGE #3 - DUPLICATE HASHES

- User registers => create a hash (5 hexadecimal characters)
- SQL procedure writing to tables and generating hash

Problem

- Some requests took 30s and got cut off by Heroku
- SQL queries stuck in Postgres (no timeouts configured)
- Out of hashes after 1M+ ($16^5 = 1\,048\,576$ unique combinations)

Solution

- Increase the hash length 🙋

```
while(!foundHash) {  
    foundHash = generateHash()  
}
```

THANK YOU!

Jozef Cipa / @jozefcipa

STRV

QUESTIONS

STRV