prototype.berlin
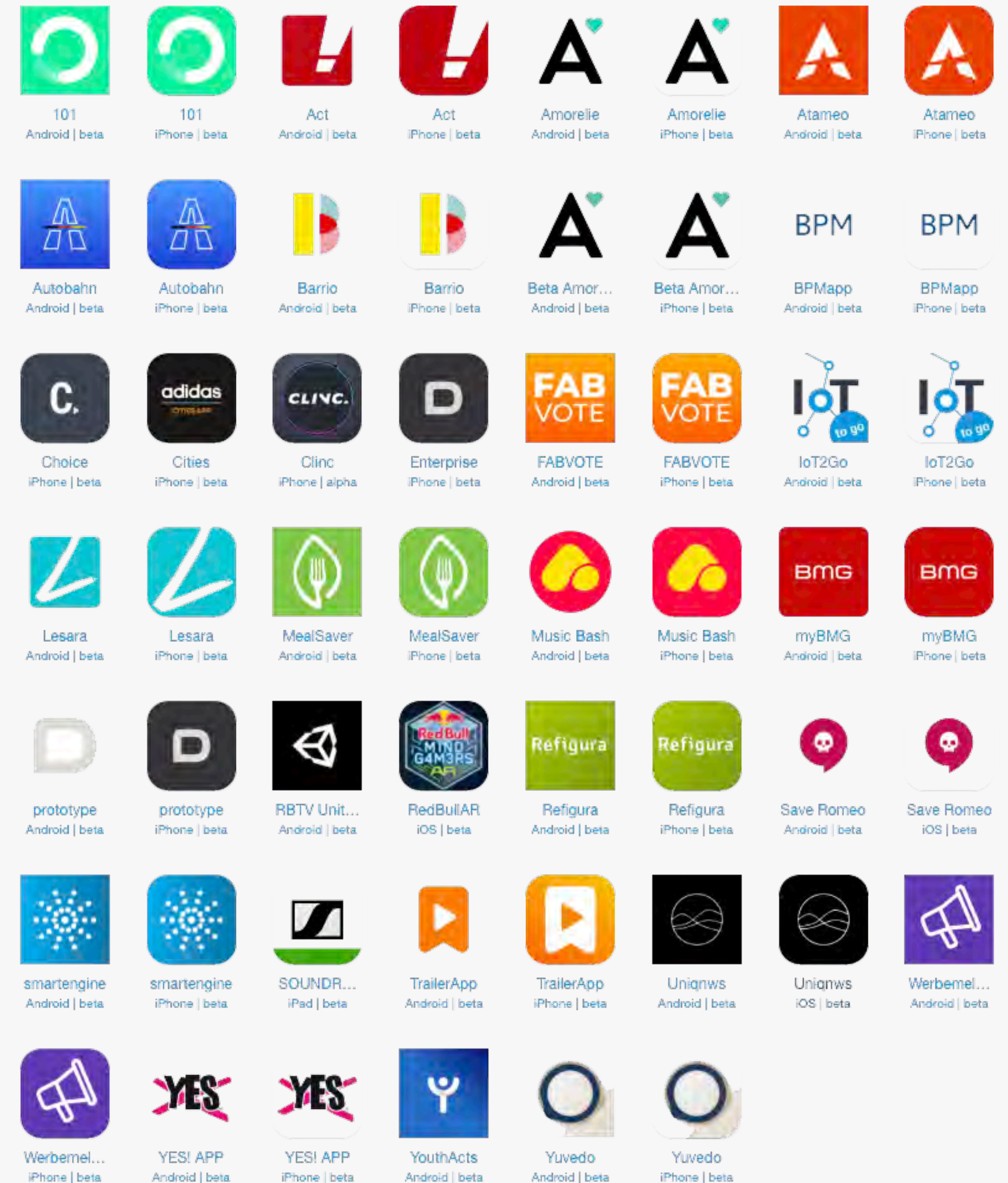## app development & design

IONIC MEETUP

September 2019

Ionic case study

Deployment as native apps and PWA from the same codebase

We have built 26 Ionic Apps over the years

As well as a handful of PWAs. With Yuvedo, we used Ionic for the first time to deploy the same code base as natives apps and to the web.

Yuvedo is Freeletics for Parkinson's patients

YUVEDO is by people with Parkinson's for people with Parkinson's. We want less tremor and less rigor, sleep better and speak louder. We want to accompany our children to the prom, play catch with our grandchildren and travel with our families. We want to keep our freedom or win back. That's why there is YUVEDO. Together we will reduce our symptoms and live a better life longer.

The project is sponsored by the AOK.

Features

Courses

- ✓ Courses with different exercises
- ✓ Video exercises
- ✓ Team view with gamification
- ✓ Regular meetings via Video chat

Chat and user profiles

- ✓ Real time chat with Websockets
- ✓ One-to-one chats
- ✓ Group chats
- ✓ Rich but anonymous user profiles

Statistics

- ✓ Track the Yuvedo score
- ✓ Track exercise ratings
- ✓ See past courses
- ✓ See correlations between exercises and the Yuvedo score

BACKGROUND

- **Why Ionic?**
  One codebase, all the relevant platforms
  (iOS, Android, Web)

- **Project-specific challenges:**
  - Patients with tremor need to be able to use the app
  - Helping a startup get off the ground means
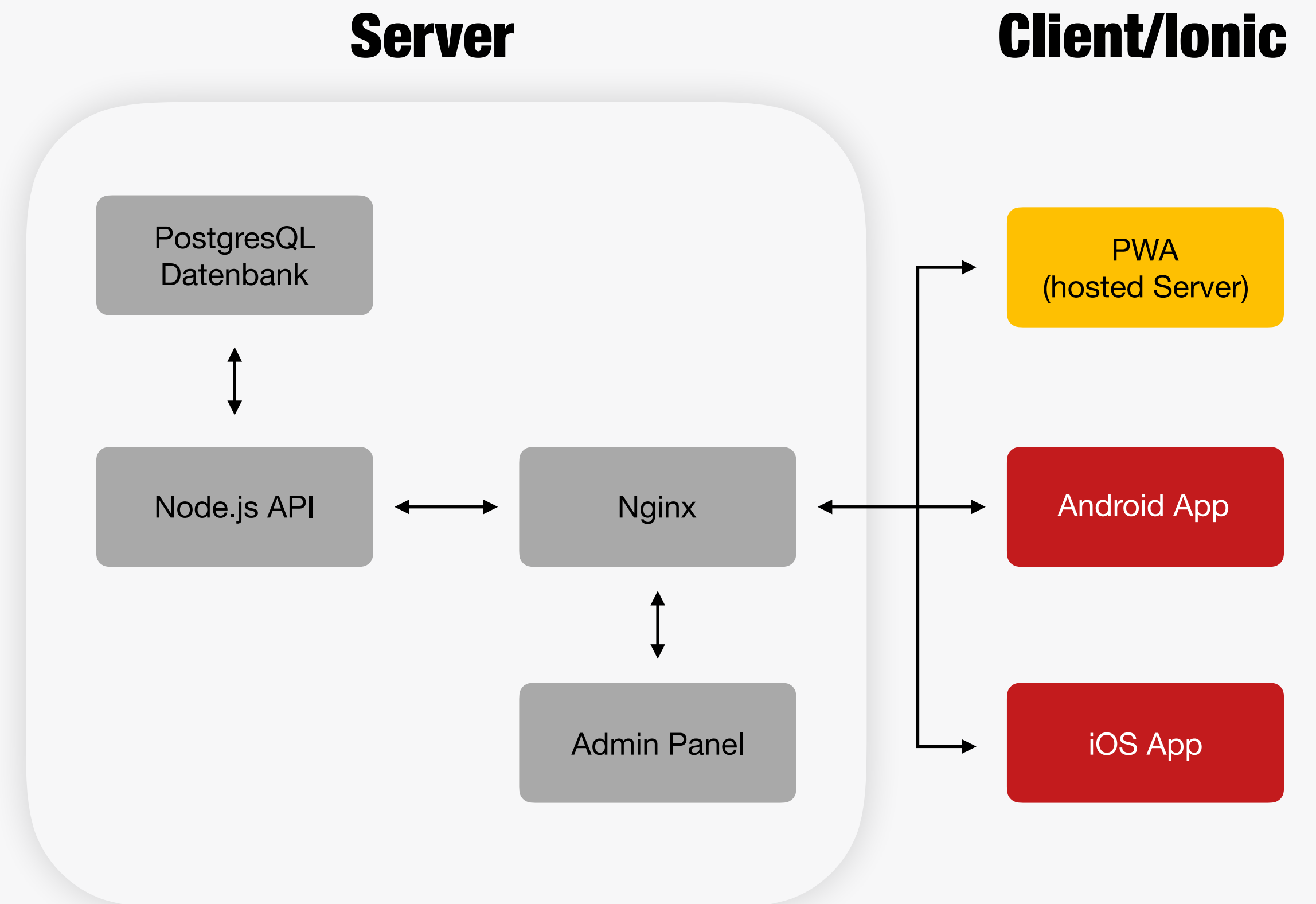   that requirements will definitely change, and fast

Inhalt

Deployment was easy, but requires optimization

- The stack: Ionic 3, Angular 5 (released before Ionic 4)

- Basic deployment easy as pie, just build and upload the www-folder somewhere

- Out of the box, loading times for first load are not optimal
  ~6 MB and 33 s until ready

- With caching, lazy loading and gzip compression, loading times of Ionic can be alright
 ~2 s and <1 MB until ready

Server

Client/Ionic

PostgresQL
Datenbank

Node.js API

Nginx

Admin Panel

PWA
(hosted Server)

Android App

iOS App

Overcoming platform-specific design-challenges

- Tab navigation works well in mobile apps but not on desktop

- Header design in apps implies ?

- Some navigation patterns differ, e.g. header back buttons
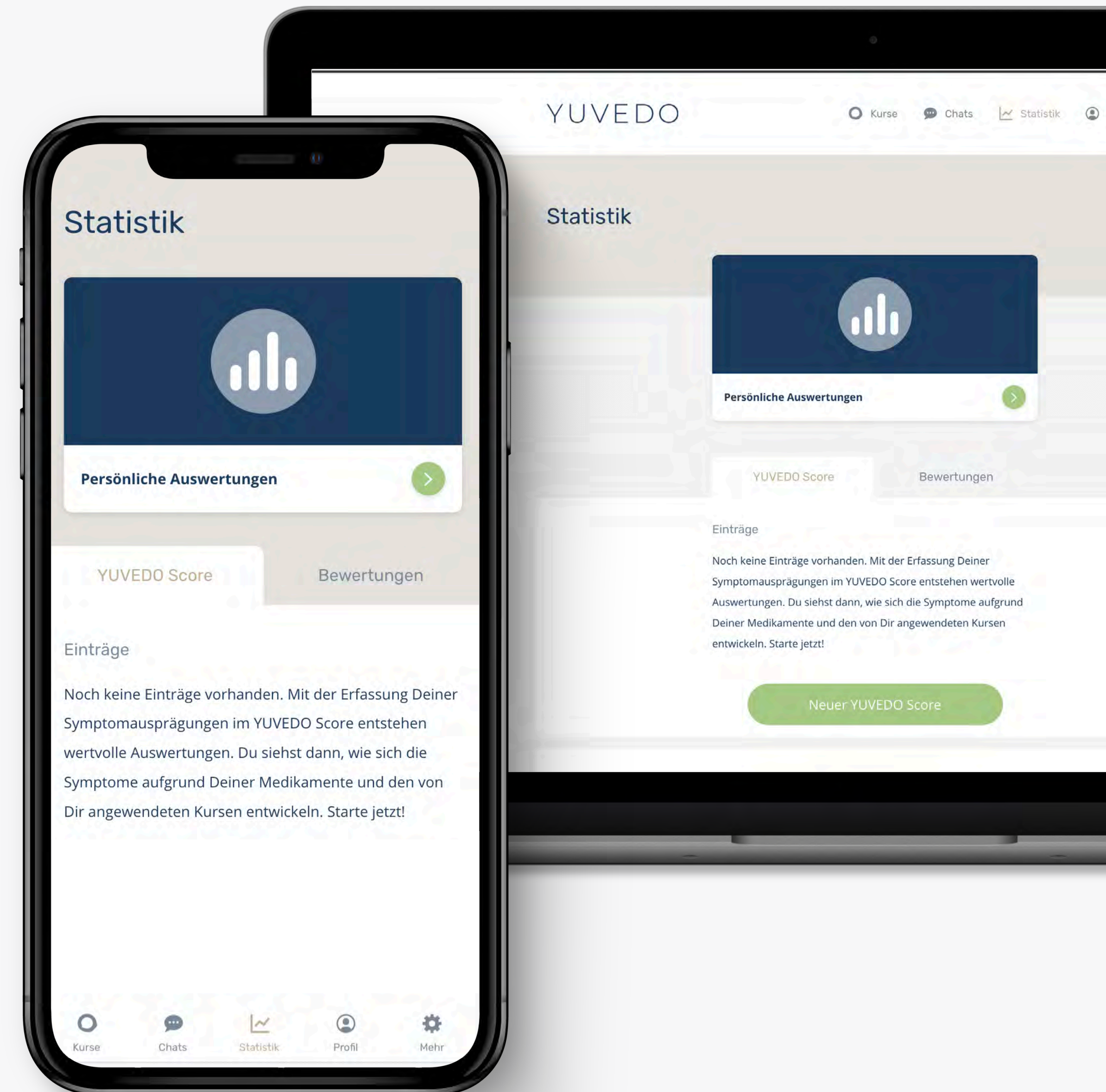
Most Ionic components worked just great on bigger screens

- ion-grid is awesome

- ion-slides is very adaptable

- AlertController, ModalController and ToastController work very well

- ion-datetime felt weird, we replaced it on desktop with a number input (only year was required)

- ion-select feels quirky on desktop, users are not used to confirm single option selections
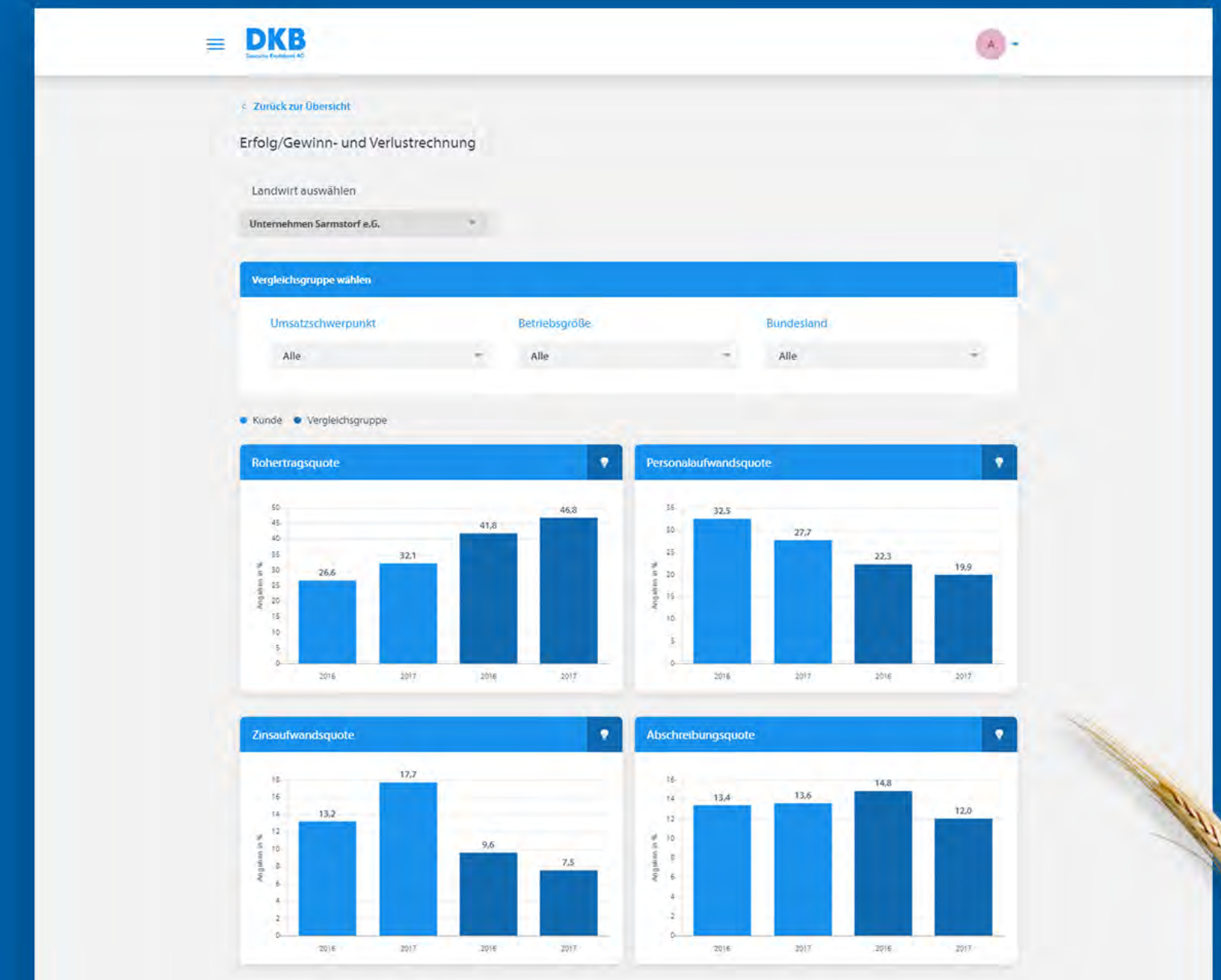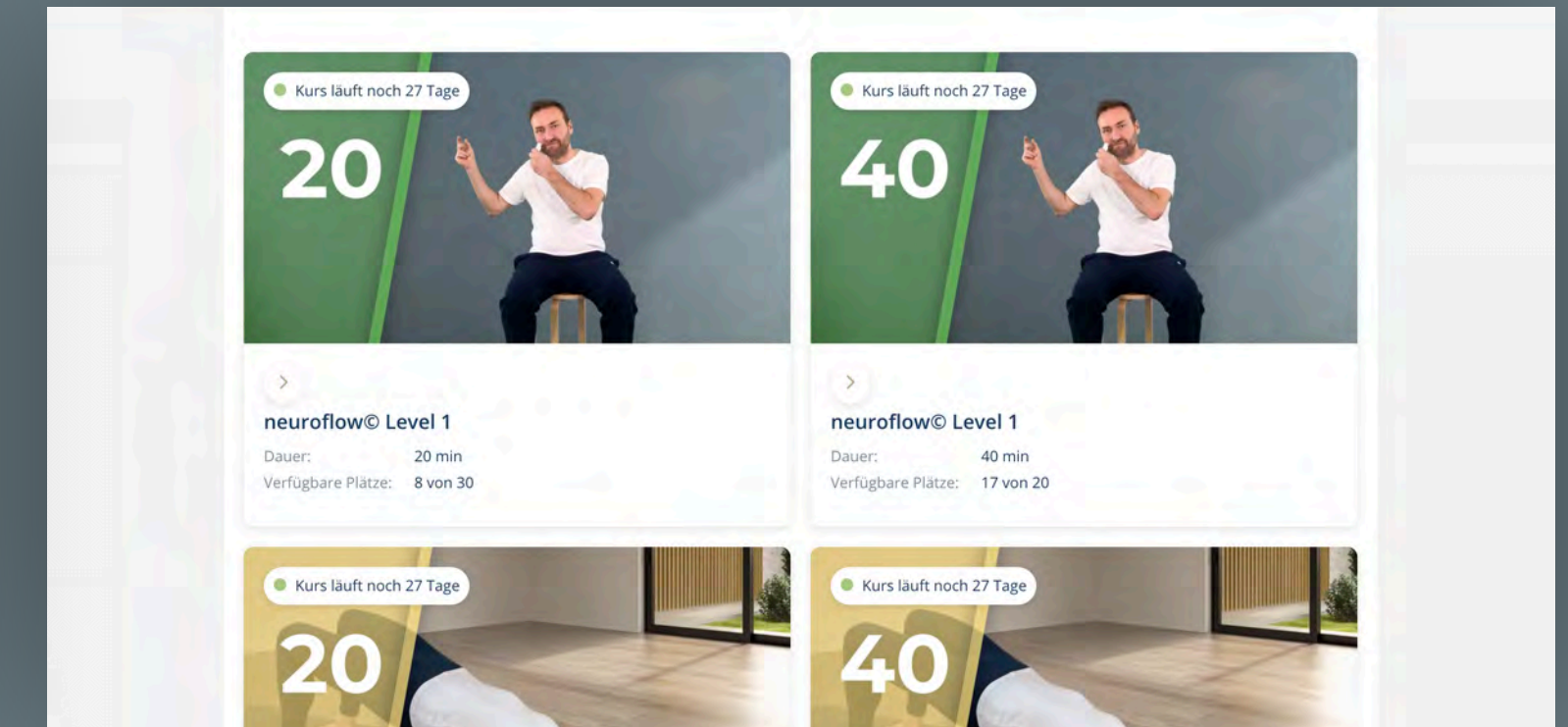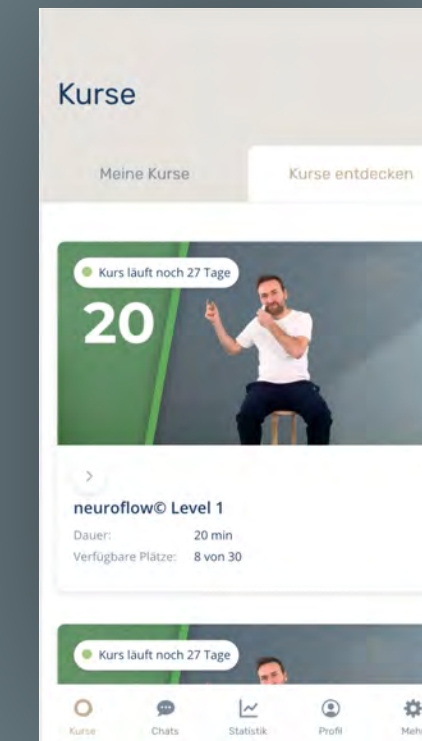
Payment is the major difference between the platforms

- We've used Stripe and it's lit 🔥

- We've tried the Netflix approach:
  Registration and Payment only in the Web App

- That didn't work with Apple this time 😬

- Because Stripe can't be used on iOS, this part is the one
 with the most differences across platforms
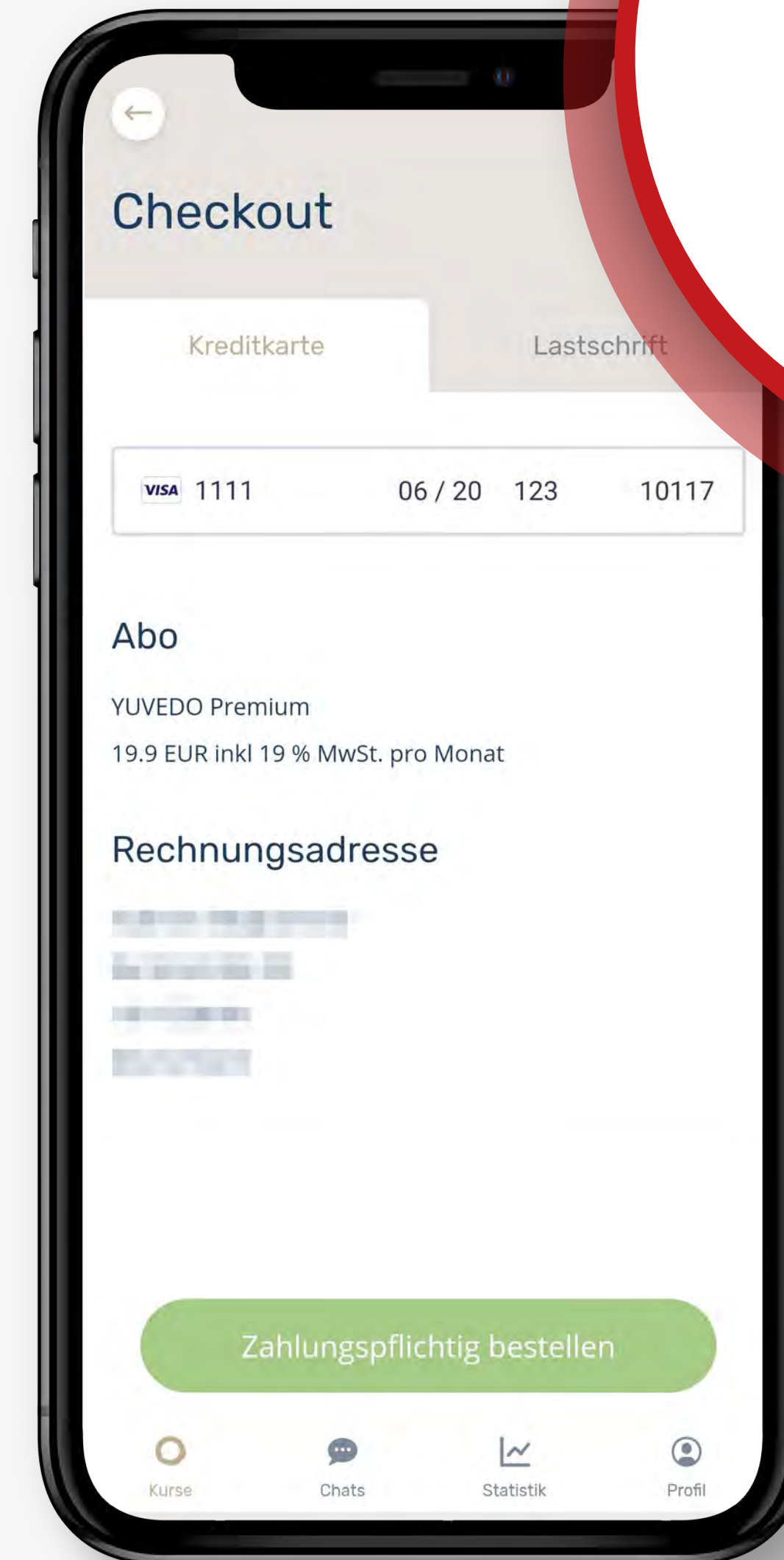
There's always something,
but: more platforms, more issues

- Video playback across platforms was finicky (looking at you, Android) but in the end, it worked decent with MP4 streams from Vimeo

- We like to use the Appinfo plugin and had to mock it for the web

- Surprisingly, Edge isn't a problem at all
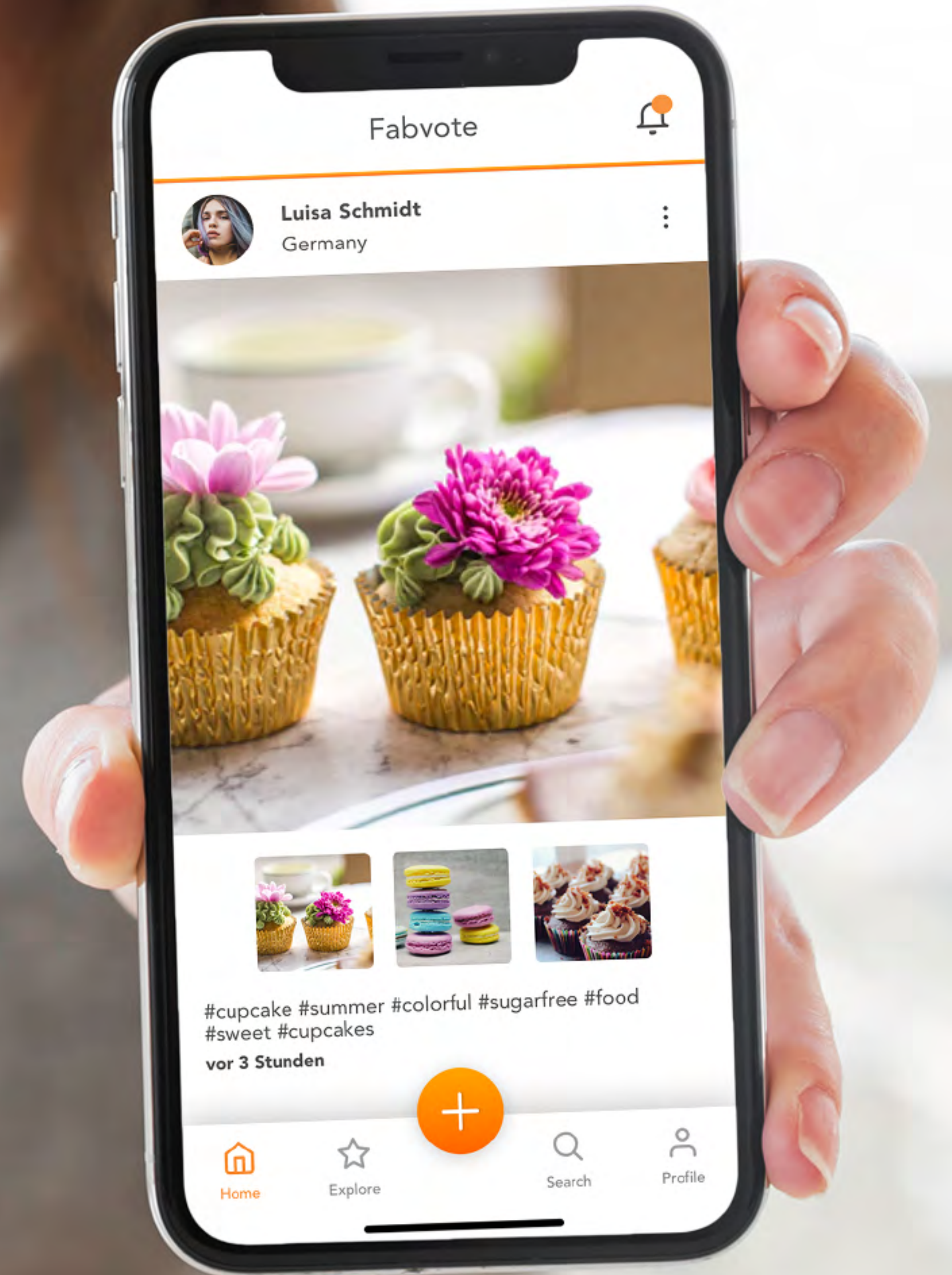
In summary: We would do it again, no doubt

- We could iterate the product quickly across three platforms

- Most cross-platform issues were minor and/or quickly overcome

- Continuous Deployment to the web is definitely less error prone than Cordova (Google Firebase, anyone?)

- Ionic has quite a big footprint out of the box, so some thought should be put into cost vs. benefits
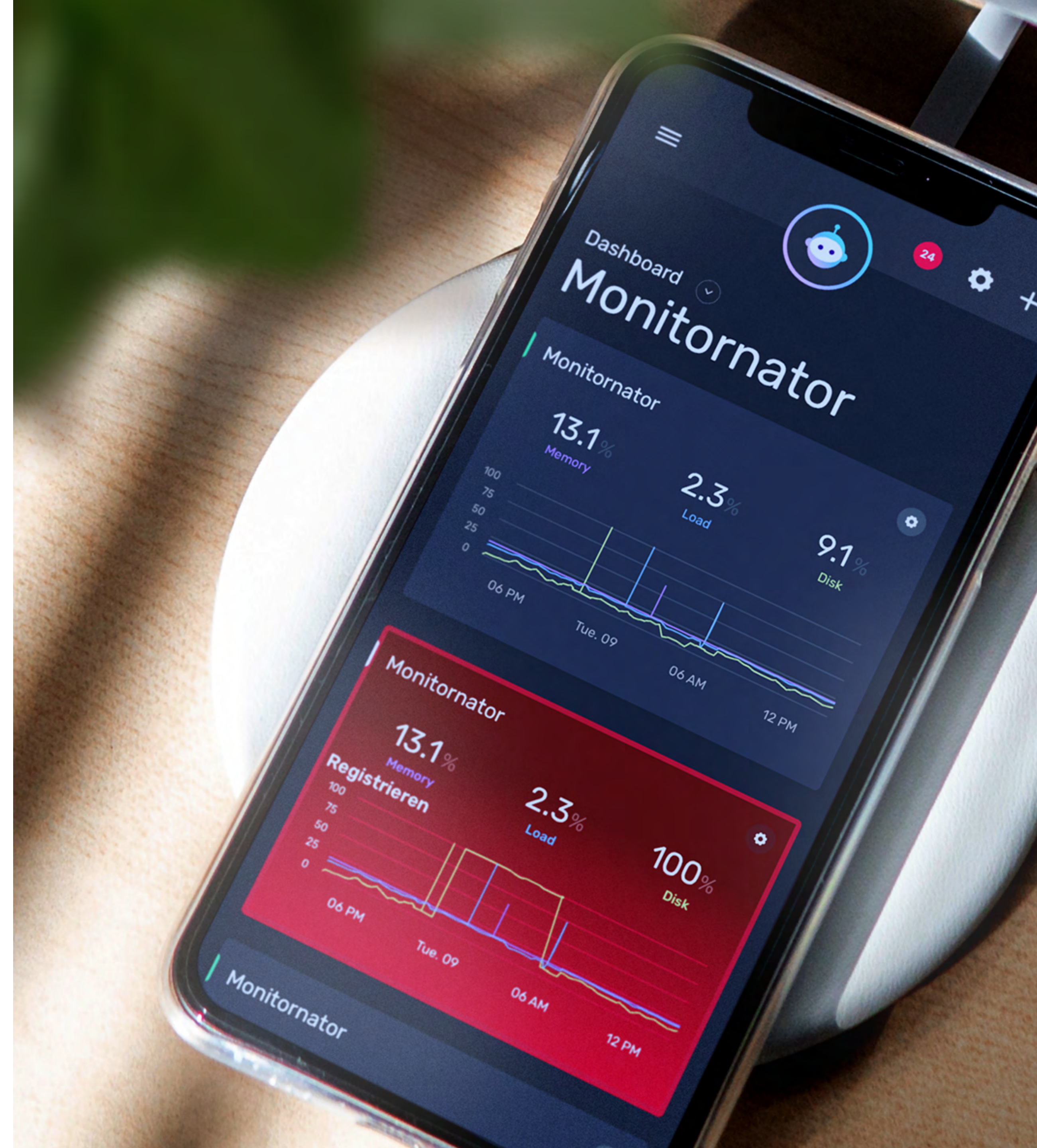
QUESTIONS?

Don't hesitate to ask 🤗

prototype.berlin
app development & design

prototype.berlin GmbH
Oranienburger Straße 26
10117 Berlin

welcome@prototype.berlin
+49 30 21950860

Ruben Gänsler
Co-Founder & Digital Strategie

ruben@prototype.berlin