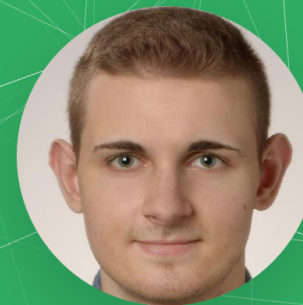




Philipp Matthes

<https://philippmatth.es/>

philipp.matthes@tu-dresden.de



Felix Kästner

<https://www.felix-kaestner.com/>

felix.kaestner@mailbox.tu-dresden.de



Here is the core idea.

You visit a restaurant, café or bar
and *order online.*





Scan the
QR Code to
directly
start ordering!

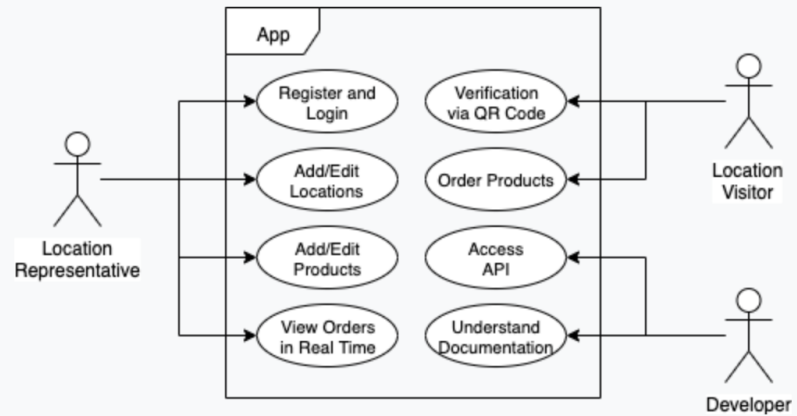
I AM A LOCATION

ORDERED.ONLINE

IMPRINT
PRIVACY POLICY
TERMS OF USE



Use Cases



Challenges

- How to handle interrupted connections during ordering?

Solution: store actions server side and cache, where possible.

- How to balance feature quantity and energy consumption?

Solution: identify power intensive tasks through profiling and mitigate usage of these.

- How do we ensure optimal usability?

Solution: recruit test users and evaluate peer feedback.

- How do we reduce development effort and ensure quality?

Solutions: Develop interoperably in the frontend with React Native.

Build on the base of micro services to choose a suitable technology for each functionality.

Use Docker to reduce deployment efforts. Use continuous integration to perform automatable tasks (e.g. building the documentation).



Briefly: use case specific challenges

- How can the location representative confirm, that an order was placed by the one sitting at the table?
- How can a user confirm, that he orders at the right location?
- How can the location representative confirm, that the user paid?

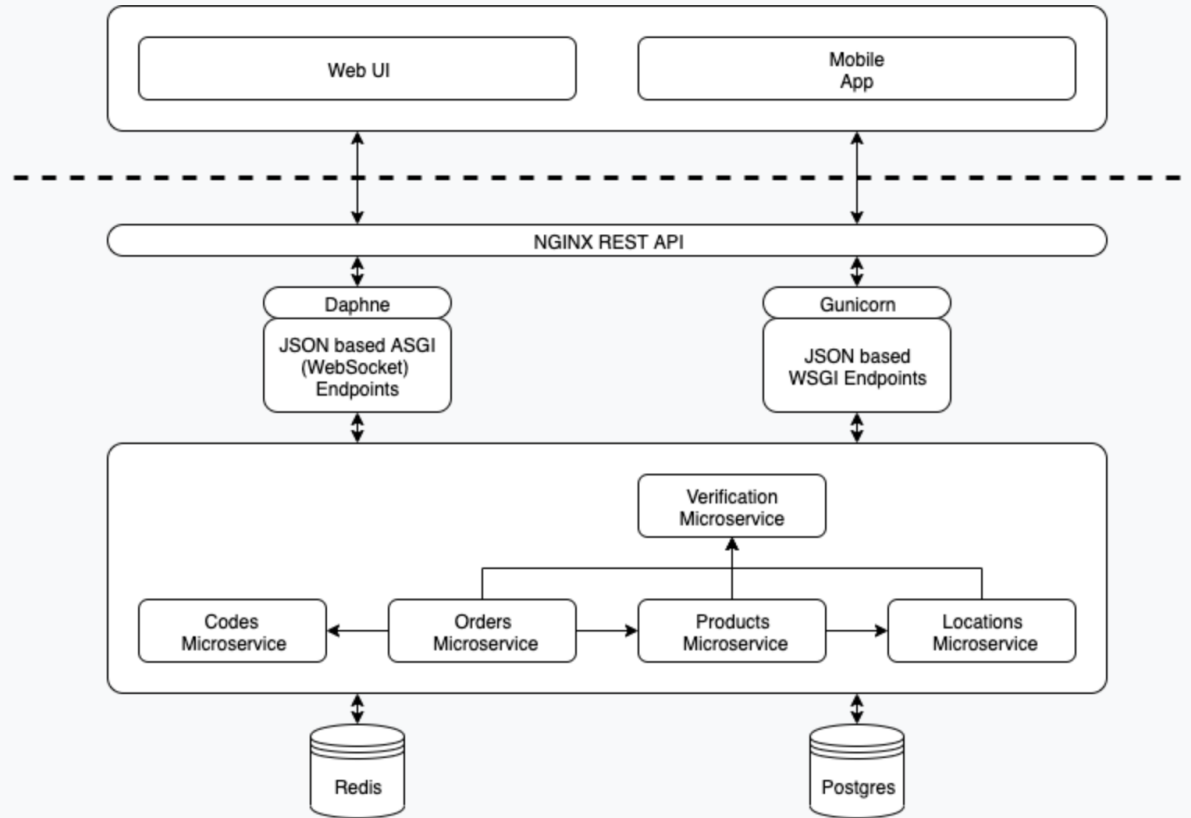


Adaption in our app

- Allow autonomous changes of the location representative's order panel through WebSocket channels.
- Handle the scan of the location representative's QR code appropriately, i.e. allow product ordering.
- Allow the location representative to close the order session and handle this in the client.



Project Structure



Powered by...



NGINX



redis

Expo

React Native

django

PostgreSQL



Schedule	Event
Nov 1 2019 to Nov 15 2019	Finish backend (✓), build prototype with base functionalities
Nov 15 2019 to Dec 6 2019	Implement all functionalities in the prototype, deploy web app
Dec 6 2019 to Dec 11 2019	Prepare prototype for peer review, i.e. submit App to Testflight (iOS) and create a test track (Android)
Dec 11 2019 to Jan 24 2019	Reiterate and improve app based on peer feedback, implement additional features, make it production ready. Increase scalability.
Jan 24 2019 to Jan 31 2019	Go into production, i.e. publish to App Store and Play Store

