

## README

We have built a solution for the mobility vertical that gives users access to a range of mobility services via a single app.

- Trains
- Buses
- Trams
- Ferries
- Ropeways
- Flights
- Car rental
- Bike rental
- Car sharing services
- And more...

The system is a combination of node server that connects to the Stellar blockchain and an Android app for the end user.

Node Server - <https://github.com/kimeoapp/kaoru-server>

Android App- <https://github.com/kimeoapp/kaoru-android>

Android app

<https://sbhack-video-submission.s3.eu-central-1.amazonaws.com/team11/kar.apk>

### How it Works:

- We have used the train service as sample use case.
- The user scans the QR code at the beginning of the service, the system deducts a deposit, defined the Config JSON, from the **User wallet** and deposits it to a shared **Trip wallet**. (*Boarding Station*)  
Boarding Station Sample QR (Scan with demo app to test)



- The user scans the QR code at the end of the service, the system sends the payable fare, defined in the Config JSON, to a Shared **Service wallet** and the balance back to the **User wallet**. (Deboarding station)  
Deboarding Station Sample QR (Scan with demo app to test)



- If the user uses the service multiple times during a billing cycle, defined in the Config JSON, the Shared Service wallet keeps accumulating the fare.
- At the end of the billing cycle the discount policy, available in the Config JSON, is applied (10% discount on 15+ trips). The discounted fare is transferred to the **Service provider wallet** and the balance is transferred back to the **User wallet**.
- The system also credits the user with Green and Black credits for use of environment friendly services. The credits per service are also available in the JSON Config.

- There could be a marketplace in the future that allows people to offset their Black credits by paying the holders of Green credits.

**Scalability:**

The system has been designed to take JSON object as an input config so a range of services can be added by adding different config parameters.

The app pulls the config based on the QR code.

The gateway to the app is just a QR code, the app pulls the config file for the service from the server when a QR code is scanned.

The QR codes can be printed and applied to service delivery centers, each QR code will have the config file address and the location info.