Authors: Floran NARENJI-SHESHKALANI (643166) & Jean-Marcellin TRUONG (643357)

1 FEATURESET

This ski ticket can hold both value and subscription. The former is stored as a decrementing number of rides. For the later, an unactivated subscription is stored as a number of hours, while an active subscription is stored as an expiry date. Subscription will be used before rides (e.g. having a 5-days subscription on going means that the number of rides won't decrement).

This design is simplistic, and does not support advanced features such as cash on the ticket, zones or pay-per-hour subscriptions.

With the goal here being NFC design rather than skiing resort ticket specification, it was felt that the provided features were sufficient for a proof of concept. Furthermore, the current design uses less than half of the storage space, and already provides a tear-proof, tamper-proof way, eventually fraud-proof (as in 'eventually consistent') way of storing offline information on the card. It should therefore be quite straightforward to extend the features for more complex use cases.

The ticket also provides pass back protection (e.g. giving the ticket to the person behind you), with a one minute timer.

2 IMPLEMENTATION

The prototype runs on the PC platform, instead of the default Android. It was felt that developping on a PC provided a faster edit/build/debug cycle and also had good support for unit testing. With the provided NFC library being in Java, it was elected to write the prototype in Scala. Scala was chosen because it is generally a more modern, elegant and fun language compared to Java.

For manipulating binary data, we use the `scodec` library. This library offers an algebra for mapping Scala data structure to binary format by usage of codecs. A lot of useful codecs are provided by default, however the use cases at hand required some hand-made codecs (e.g. for MAC, offset date time storage...). Because of `scodec`, it is harder to work out where each piece of information is (requires some mental computations), but the size of the whole data structure is known which is sufficient.

`scodec` is interesting because it works, by design, with a bit granularity, thereby resulting in a denser output, while avoiding hand-written bit manipulation. This is obviously advantageous in a space-constrained environment such as an NFC card.

For unit testing, `ScalaTest`, the standard unit testing library for Scala, is used.

3 MEMORY STRUCTURE

The first usable page is page 4.