GOVELO

Prototype Brief

Brief

The prototype needs to fulfill the following

1) Fulfill the use cases stated in this document, this is essential.
2) Contain the additional hardware for future updates to the prototype, to be discussed and agreed.
3) Be easy for us to update/flash the hardware without having to ship the unit back to Taiwan
4) Be possible to realise manufacturing within the Industrial design envelope.

Index

Prototype Use Case

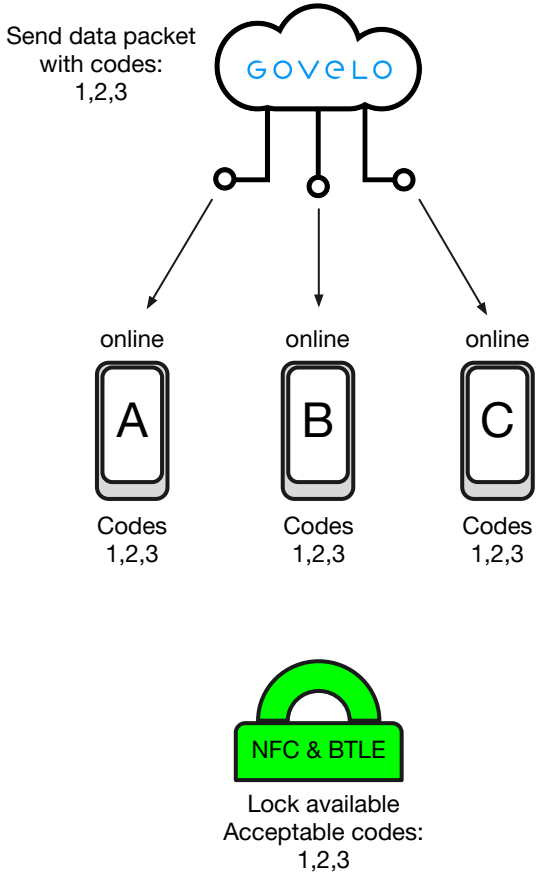
The prototype lock and basic app must satisfy the following use case and edge cases.

1: Lock access codes

The Cloud server sends out data packets to our app containing unlock codes for the locks. These data packets are geofenced to keep their size small. In the example there are only 3 codes to keep things simple in this example.

The smartlock can store up to 3 access codes in its memory. In this starting condition the 3 codes that the smartlock and cloud server have are the same.

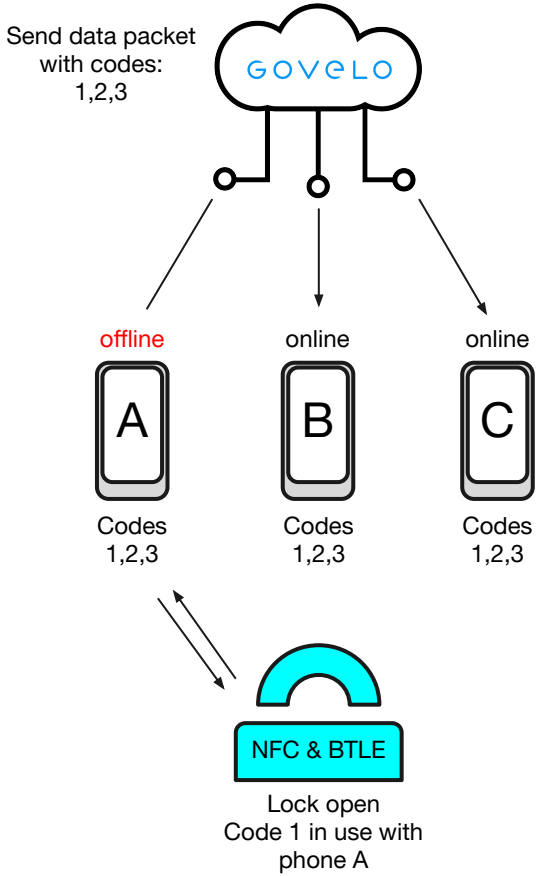
Users A, B and C all receive the data packet from the cloud server containing the access codes for the lock in their local area.



2: User Unlocking

User A walks into the underground car park where the lock is located, their phone is not longer connected to the internet.

User A connects to the smartlock via Bluetooth Low Energy using the NFC to authenticate access. User A's phone then transmits its access codes and unique phone ID to the smartlock via Bluetooth. The lock selects one of the valid codes, code "1" in this case, and links it to the phone's ID. The smartlock now unlocks with an audio cue "beep". While unlocked the lock's LEDs are softly pulsing blue.



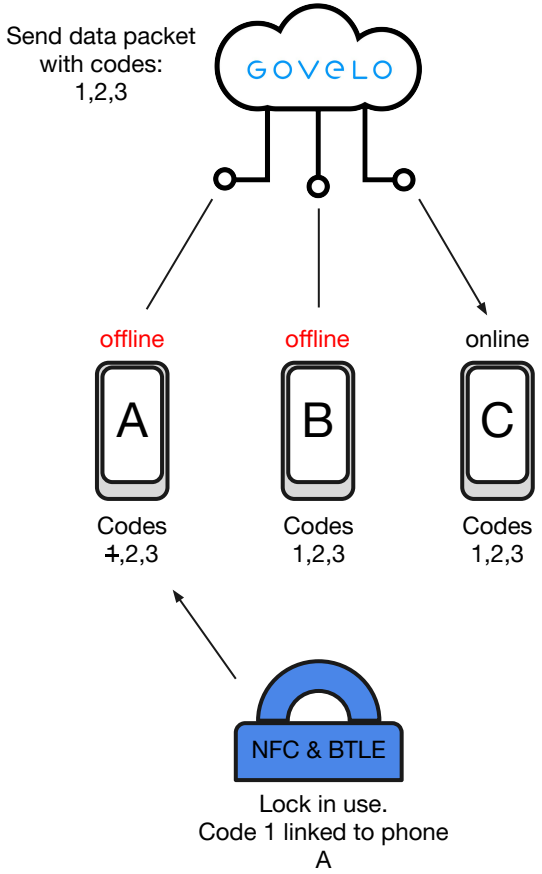
3: User Locking

User A now uses the lock's cable to go through their bike and wheels before plugging the cable back into the lock.

When the smartlock senses the cable being re-inserted it will activate the locking pin, emit a locking audio "beep" and turn its LEDs to a constant blue (see edge case #1 for when the lock cannot successfully activate the locking pin). The smartlock sends a "lock successful" message back to the phone and the app updates its data packet to send back to the cloud server when re-connected to the internet.

Now the smartlock is in a locked state where the only code it will accept to unlock will be code one transmitted by phone A. All other unlock requests are ignored.

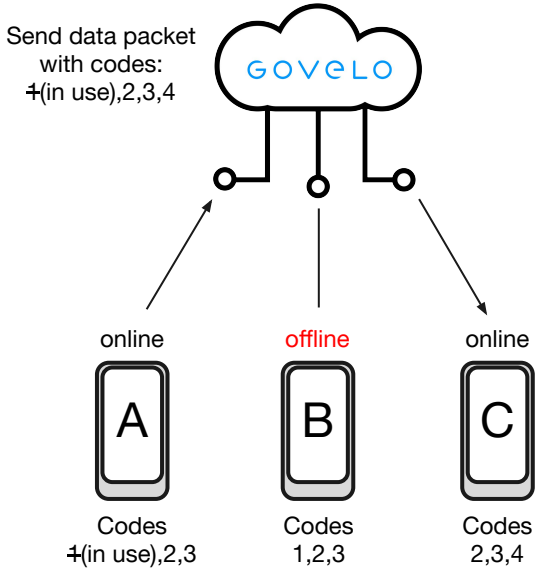
While this has been happening, user B's phone has gone offline and is no longer getting code updates from the cloud server.



4: updating the server codes

User A has now left the underground parking and their phone reconnects to the internet, transmitting its data packet back to our cloud server.

The cloud server now knows that code 1 is no longer available and generates a new code to replace it, this is code 4. The server then starts distributing the new data packet with the updated code list to users who are connected such as user C. User B is still offline and therefore is has the older code packet.

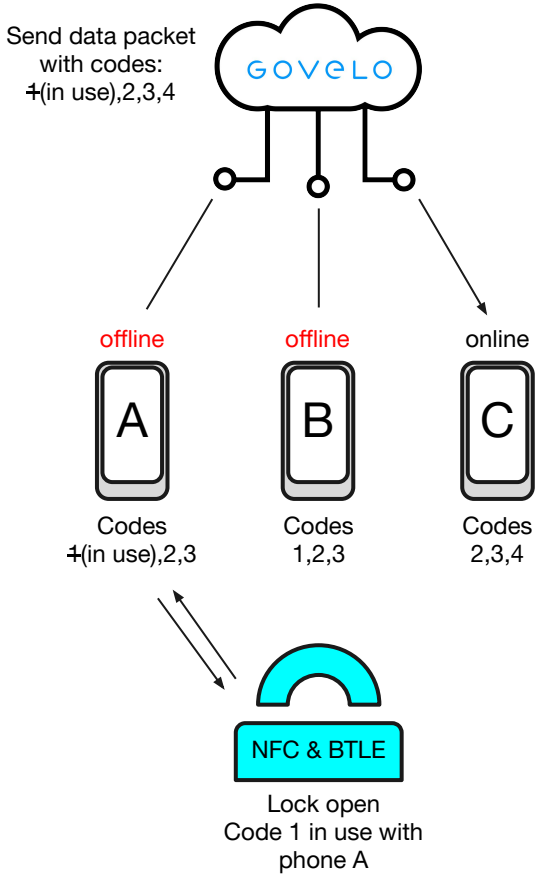


Lock in use.
Code 1 linked to phone A

5: Return and reset

User A is back in the underground parking to unlock their bike. User A connects to the smartlock via Bluetooth Low Energy using the NFC to authenticate access. User A's phone then transmits its access codes and unique phone ID to the smartlock via Bluetooth. The smarlock recognises code 1 and phone ID. The smartlock now unlocks with an audio cue "beep" and the lock's LEDs are softly pulsing blue.

User A takes their bike and plugs the cable back into the lock as requested by the app. The smartlock senses the cable being re-inserted it will activate the locking pin, emit a locking audio "beep" and turn its LEDs to a constant green (see edge case #1 for when the lock cannot successfully lock). The smartlock sends a "lock successful" message back to the phone and the app updates its data packet to send back to the cloud server when re-connected to the internet.

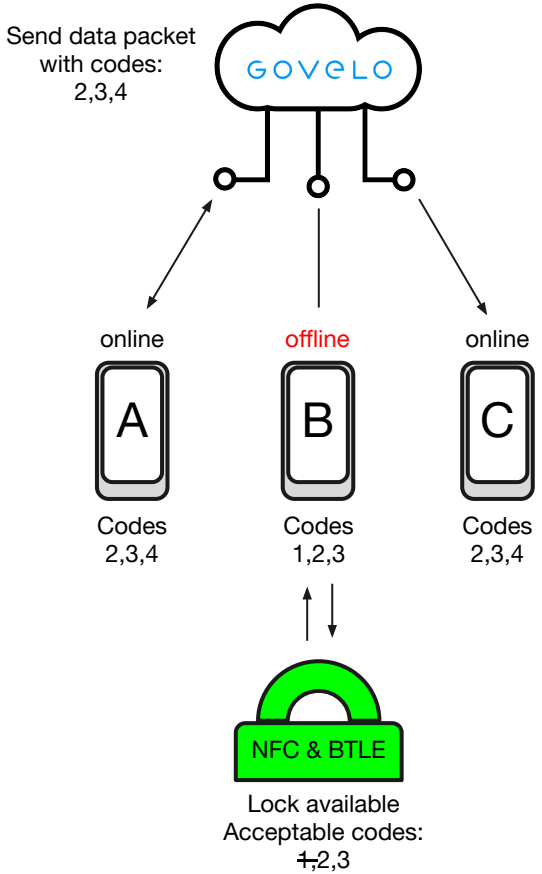


6: Next user access

Once user A's phone reconnects to the internet the cloud server is updated to confirm the transaction and use of code 1 is complete. The transaction is the logged and billed to the customer's account and code 1 deleted. The server updates user A's phone with the updated codes.

User B who has been offline this entire time is now in the underground parking when he open the app to use a lock. The app cannot connect to the internet to get the lastest codes so it checks the timestamp of the code packet it has, the timestamp is less the 48 hours old so the app tried to connect to the lock with the codes it has (see edge case #2 if timestamp is too old).

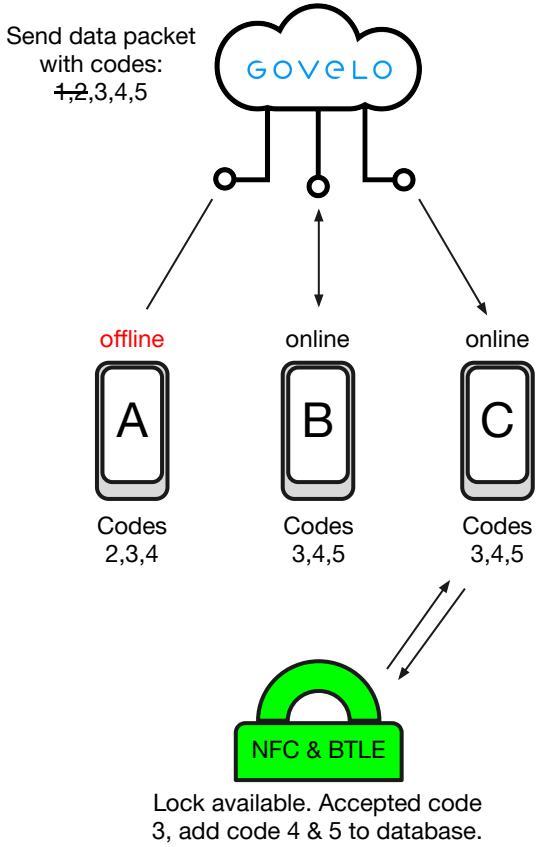
The smartlock accepts code 2 and logs user B's phone ID, and the whole process starts again.



7: smarlock code update

Now that user B has collected their bike and reset the lock, the smartlock only has one code left, 3.

User C arrives, his phone has an updated data packet with new lock codes. Same as the previous 2 users, user C connects to the lock and is able to authenticate with code 3. The smarlock sees the two newer codes 4 & 5 and updates itself with those codes for future users. (See edge case #2 is the user does not have new codes for the lock).



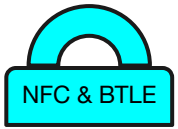
GOVELO

Edge cases #1

If the smartlock cannot successfully lock after it has been opened it will continue to pulse blue for X minutes.

If the lock detects that the user is walking away from the open smartlock (via Bluetooth signal strength) it will start flashing red and the app will warn the user that the lock is not properly closed. The lock will also make loud warning sounds to make sure the user knows it is not properly locked.

If after X minutes the lock has still not been properly closed it will turn flashing red and be unusable until it has been properly closed first. New users wishing to use the lock will be notified via their app to first close the lock so that it can reset and self test before it is safe for them to use.



Lock still not secured. Send notification to phone.

Signal weakening. Send notification to phone, start flashing red!

Edge cases #2

If a user's data packet is more than 48hr old the app will tell the user to reconnect to the internet so that it can update its access codes before they can use our smartlock. This is true even if the phone's outdated data packet has a code that the lock is able to accept.

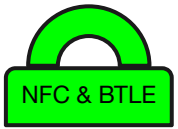
If a user is using a data packet that is less than 48hr but the codes do not match the smartlock's database then the lock sends a "out of sync" signal and the phone will ask the user to connect to the internet to update their access codes.

When a smartlock only has one valid code left in its database it will only accept to be used by a phone whose data packet is able to provide it with new future codes. This is to prevent the lock from running out of access codes and remaining stuck in a locked state.

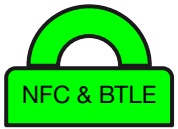
Code 1,2,3
>48hr old!
Update required!

Code 1,2,3
<48hr old.

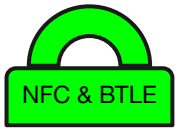
Code 1,2,3
<48hr old.



Lock available
Acceptable codes:
3,4,5



Lock available
Acceptable codes:
7,8,9
Send signal to phone
"Out of Sync!"



Lock available
Acceptable codes:
~~1,2~~,3
Send signal to phone
"Smartlock needs new codes!"

Additional Hardware

Hardware that is not part of the current prototype use case, but that we would like to have in order to develop further use cases later.

1: Accelerometer

We want to be able to detect vibrations so that in the future we can get the smartlock to monitor its environment for potential vandalism / theft.

2: Decent BTLE chip

BTLE powerful enough to ping simple data about the lock's status via something like the PXP protocol. Ideally no less than 2m range, the more range the better.

GOVELO

Industrial Design & Manufacturing

To give you an idea of what the final product package will be and how we think it could be manufactured in order to get a rough idea of unit cost.

Current design:

The idea behind the current design is to make the lock look modern yet strong and secure while keeping manufacturing and tooling costs relatively low. This design also allows for secured yet easy maintenance and replacement of the batteries.

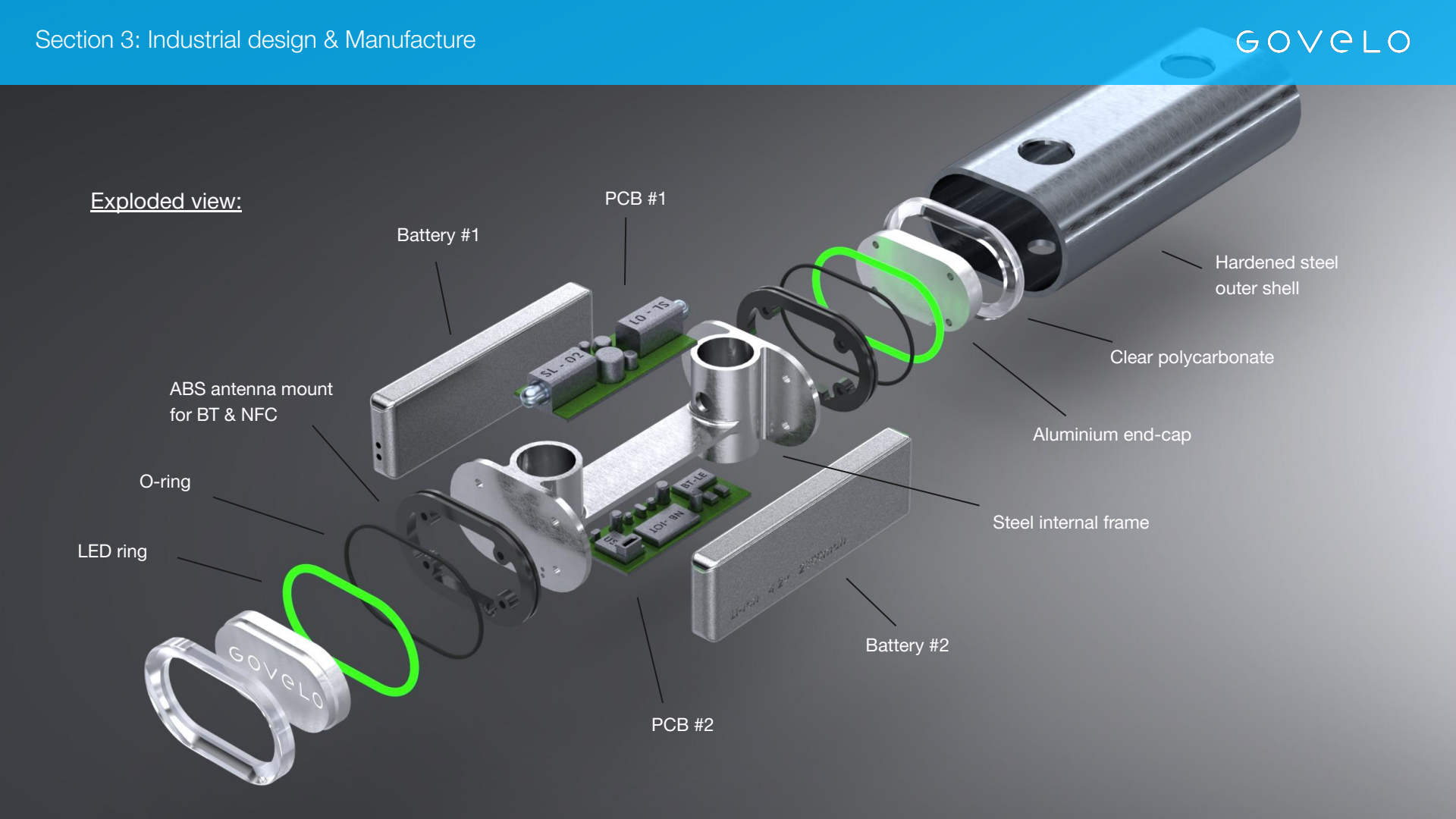
The lock's dimensions are not set in stone so if the hardware needs more space we can easily modify the design to accommodate it.

GOVELO

Design Features:

- Up to 5600mah Li-ion battery space (2x2800)
- 2x PCB 20mm*140mm max volume
- 5mm hardened stainless steel outer shell
- 22mm dia lock bore
- 10mm dia locking pins
- 2.5m*20mm plastified steel lock cable
- 40mm*80mm*175mm lock outer dimension

GOVELO


Exploded view:
Battery #1
PCB #1
Hardened steel outer shell
Clear polycarbonate
ABS antenna mount for BT & NFC
Aluminium end-cap
O-ring
Steel internal frame
LED ring
Battery #2
PCB #2
GOVELO

Assumed manufacturing techniques for
<u>non-stock lock body components:</u>

A - Outer shell - Stock hardened steel oval tube, cut to length and 4 holes punched/cut/drilled. Possibly laser etched logos / graphics.

B - Inner frame - 2 sections of stock steel tube cut to length and holes drilled + 3 plates of steel cut to spec. The 5 pieces are then accurately welded together to form the complete frame

C - ABS antenna & o-ring mount - CNC machined from block

D - Aluminium end-cap - CNC machined / cut from sheet

E - Clear Polycarbonate - CNC machined / cut from sheet