École Polytechnique Fédérale de Lausanne

# Anonymous Proof-of-Presence Wallet

by Karim Kabbani

## Project Report

Approved by the Examining Committee:

Prof. Bryan Ford
Thesis Advisor

Expert Reviewer
External Expert

Thesis Supervisor
Thesis Supervisor

EPFL IC IINFCOM HEXHIVE
BC 160 (Bâtiment BC)
Station 14
CH-1015 Lausanne

June 11, 2021

## Introduction

The project I was assigned to is the PoP wallet on the Android front-end. At the beginning of the semester Filippo and I worked together on a very simple implementation of a wallet. Very soon in the semester, the Java team was asked to complete the roll-call implementation as it was not yet functional. I decided to take care of this task with the goal of being able to fully run roll-calls on the Android application during PoP parties. Being from the wallet team I felt particularly involved in making sure roll-calls work correctly, because users receive PoP tokens through roll-calls, which is essential to demonstrate that the wallet actually works. Besides I was really interested in doing this part and one of the reasons is that it forced me to get a good general understanding of the codebase, which by the way was really not easy for me in the first weeks. Filippo continued working on the wallet on his side and both our tasks were very independent at this stage.

If I were to divide my contribution in parts, it would mainly be in three following (on the Java front-end):

- Completing roll-calls' implementation
- Fixing bugs including the ones encountered during PoP parties
- Wallet content implementation

## Roll-calls

I started by working on the organizer's UI and afterwards only a few things were left for attendee's UI.

### Roll-call organizer's UI

- Roll-call creation, opening and closing: I implemented the three methods that handle these in the LaoDetailViewModel and in RollCallEventCreationFragment I set up the button to create the roll-call.
- EventState: I added this enum to keep track of the current state of an Event and used it in the RollCall class.
- List of roll-calls: I added a new layout and modified the EventExpandableListViewAdapter to update the list with the correct information and buttons depending on the roll-call's state (buttons are different for attendees).
- Change in protocol: after a change by Gaurav on the proposed_start/proposed_end fields of a roll-call I adapted the corresponding classes and logic.
- Scanning of attendees QR code: the QRCodeScanningFragment already existed for scanning LAO QR code, I modified it and made it modular so that it can be used for scanning

attendees as well. I added a counter, warning and confirmation pop-ups and a button with a pop-up for closing a roll call. The scanned attendees are read in LaoDetailViewModel where they are sanity checked and added to the list.

### Roll-call attendee's UI

- Enter roll-call: I created the RollCallDetailFragment to display the generated token by the wallet.
- I put the requirement of setting up a wallet to the moment where users want to enter the roll-call. Indeed since the fragment needs to display token's public key it needs an existing wallet to be computed.
- I added an attribute to the Wallet called isSetup to know whether the seed has been initialized.

## Bug Fixing

- In the early weeks of the semester I had to fix a few bugs from the existing code regarding the events' list.
- Also at first the LAO creator was not sending a subscribe request so it didn't receive a broadcast and the roll-call could not be created. I had to short circuit the broadcast artificially in the sendPublish method of the LaoRepository to be able to test the UI and continue working. This was then fixed by our advisors.
- There were several bugs from the backend for handling the requests for the roll-calls and I shared the issues with François who worked on fixing them. Sometimes the issue came from the front-end and sometimes from the backend so we had to troubleshoot the bugs together by sharing our logs for example.
- In the LAO properties I added the channel id as a QR code so that users can subscribe. This was more a UX issue than a real bug.
- I implemented the missing code for updating the LAO name.
- The witness_signatures field we sent was incorrect in the Java app. I had to correct this in the JsonMessageGeneralSerializer class and make little changes in PublicKeySignaturePair and MessageGeneral classes.
- During the PoP parties we found some bugs for the roll-calls, there were very different kind of issues, so I spent some time on this by testing with teammates (mainly Victor and Johann). Gaurav also gave me access to the DEDIS server so that we can test the roll-calls while looking at the logs.

- I worked on fixing the inconsistencies between front-end 1 and 2 that appeared during the PoP parties, so that the roll-calls run smoothly between the two. I met with Adalsteinn a few times to troubleshoot these.

## Wallet Content

This part was about allowing a user to display his tokens for the roll-calls he attended and also for the attendees and the organizer to see the attendees' list of the past roll-calls. More details about how the wallet UI looks like can be found in the general report.

I created the following dedicated fragments, the first one belongs to HomeActivity and the others to LaoDetailActivity:

- ContentWalletFragment: it displays the list of LAOs and has a logout button to destroy the wallet.
- LaoWalletFragment: it displays the list of roll-calls of the LAO and I created a WalletListAdapter for the ListView it uses.
- RollCallTokenFragment: it displays the token the user received from the corresponding roll-call.
- AttendeesListFragment: it displays the list of the corresponding roll-call's attendees and I created an AttendeesListAdapter for its ListView.

The following modifications were needed as well:

- In LaoDetailViewModel I added a LiveData object called LaoAttendedRollCalls which keeps the list of closed roll calls that the user either attended or organized so that it can display the appropriate ones in the LaoWalletFragment.
- For the logout I had to make little changes on the public interface of the Wallet class.
- As the state of a roll-call changes, its id in the Lao object gets updated. Using a roll-call id that changes is problematic for the wallet to generate the same token for the same roll-call, because we need to always use the same id. Therefore I added a final attribute to the RollCall class called persistendId which is the id of the roll-call at its creation and used it for token generation.

## Conclusion

Overall I really enjoyed working on this project. It was the first time that I worked on an Android application and I learnt a lot from this experience. It was a heavy workload but the code base we started with was very clean (even if more comments would have been easier for the first weeks) and I could get enlightening answers when I needed. Apart from the technical side, the project

involved a good amount of teamwork and a lot of meetings. Even though this is time-consuming, it was personally very enjoyable and enriching. To finish I would like to thank Gaurav and Céline for their reactive help throughout the semester, their answers and advice during the meetings and on Slack were of great value.