Architecture design

BBW
Jasper Hu, yjhu, 4356241
Naqib Zarin, nzarin, 4384474
Ashay Somai, asomai, 4366220
Ymte Broekhuizen, yjbroekhuizen, 4246586
Luat Nguyen, tlnguyen, 4467574

May 9, 2017

Contents

1 Introduction

This document provides a high-level description of the final product and the system it uses. The end product exists out of a mobile application in the Android environment, which is developed in the Java programming language. Using the Android environment and the programming language Java, we are developing a blockchain-based web-of-trust. The idea is that you use blockchain to verify the authenticity of the user. Blockchain itself is like the name says a chain of blocks and each block contains the information to verify a user, normally this is a public key.
The Android environment is an open-source platform and operating system, which is developed by Google. This operating system is available for many devices. However, we are focusing on the mobile devices.
Java is an object-oriented programming language. A Java program consists of many different Java classes. In these classes, the functionality of the program is written.

Hereafter, the paper will be composed of the following sections. The next section will be the design goals of this project. In the third part, there will be more information about the software architecture views of the project. Specifically, these views are Subsystem decomposition, Hardware/ software mapping, Persistent data management and Concurrency.

1.1 Design goals

Since the concept design goals are too large, it is easier to split it in six different aspects: availability, manageability, performance, reliability, scalability, and securability. Consequently, these six aspects are elaborated in the following sections.

1.2 Availability

The idea is that there will be a working version after every sprint week. This way, the product owner can see the features, we are working on, every week. So in case, we are going the wrong way, we can still turn to go the right way during the meeting.

1.3 Manageability

To manage our program, we use the version control program Git, and it is stored on GitHub. In case there is an available update, you can just pull the working release version from the master and use that one instead.

1.4 Performance

The program should be able to be processed on a mobile device, so the blockchains that the program is using should not be too large. It should even be limited to a specific capacity, which could be the capacity of the device. If there is a large addition of blocks to a blockchain, the program should decide which selection it should add to the blockchain and which blocks to skip.

1.5 Reliability

To ensure the reliability of our program, we use several techniques. The first technique that we are using is Test Driven Development. This technique ensures the fact that we completely understand what a specific feature should do. Since it improves the understanding of the feature, we can develop it more reliable and more efficiently. Testing is done using unit tests, integration tests and regression tests.
The next technique that we are using is Travis Continuous Integration. After every commit to our program, or in other words, after every update to our program, the program is automatically tested. Consequently, it will let you know, whether the update contains any errors.
Testing is important because it helps to find bugs and preventing unwanted behaviour.

1.6 Scalability

Since the whole program is decentralised to a mobile application, there are no limitations on the server side. The calculation of some hashes only require little processing power, so there won't be a bottleneck on the processing power. The only problem is that you have to save the blockchain to your device. So the only limit would be memory capacity of the apparatus. However, a single block uses less than 1 MB of disk space so that the limit would be colossal for the current phones. There are no other bottlenecks, so the program scales pretty well.

1.6.1 Client side vs Server side

In the future, the program could be scaled out to a server. Several components could be allocated to the server, and likewise, there are also several elements that should not be moved to the server.

The first element is the outer visible/ visual layer, or as many people call it the mobile application itself. If it were a web application, then it would be possible to run the primary application on the server and let the Android app be a layer, which only displays the web application. Although the web application has a larger usage domain than the native application, we are using the latter, since the whole application is decentralised and a native application runs smoother.

The second component will be the encoding and decoding functions of the internal part of the mobile application. The encoding and decoding functions could be scaled out to the server. The first advantage is that you move the computational requirement from the mobile device to the server. The second advantage that the probability of the functions being deciphered is decreased significantly since the encoding and decoding functions will not be saved in every application and thus every device, but on the server. The first disadvantage will be the that you need to send a request to the server to access every component and every request to the server has network latency. Thus the program will become slower and decrease in efficiency. The second disadvantage is that you need to make requests to the server. These requests could be intercepted by malicious software on the device, or it could be intercepted from the outside if the data in the request is not encrypted.

The third and last component is the persistent data storage. The advantage is that it does not require any storage on the device. The disadvantages are that it is much slower to use and that it is more complicated to save since you have to keep track of the blockchains of multiple users, instead of one.
To conclude this discussion, it is very debatable, whether scaling in or out is better.

1.7 Securability

Since the persistent data is stored in an SQLite file on the device itself, the first layer of security is the security layer of the operating system itself. The second layer is the password protected SQLite file. The third layer of protection is that the (valuable) information, like IBAN or contact information of the owner of the blockchain, in a block is saved using an encoding protocol. The latter could be any encoding protocol. However, our program is using a custom encoding and decoding protocol.
If all layers are penetrated, there is still an option to revoke a block. This way the block, containing information about another user will be invalid.

2 Software architecture views

The program is interconnected using different parts and the software architecture views elaborate on them. In the subsystem decomposition, it shows how the system is divided into subsystems. Secondly, in the hardware/software mapping, it illustrates how the hardware subsystems are connected to the software subsystems. As third, in the persistent data management, it shows how and where the data is persisted. And finally, in the concurrency part, it shows how the resources, processes are shared and how the communication works between them.

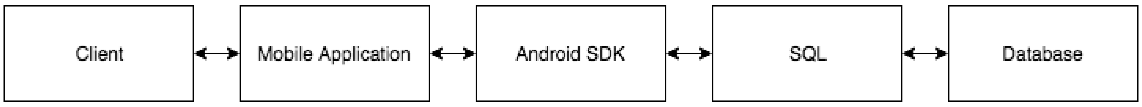

Figure 1: A graph of the interaction between the client and the database.

2.1 Subsystem decomposition

The client uses the Mobile Application to handle all requests. Or in other words, communicates with the server using the mobile application. This mobile application also processes all requests. The mobile application consequently communicates with the database using the Android SDK. Since the mobile application processes all requests by itself, there is no other communication between the database and the mobile application.

- Android SDK
  Since it is not possible to directly communicate from a mobile application, there is also middleware involved to make the communication possible.
- Database
  Since the data needs to be persisted, a database has been used. The client uses the persisted data from the database to handle any other requests.
- Mobile Application
  The mobile application ensures that the requests from the clients are made possible. This same application also processes the requests from the clients by processing the persisted data from the database.

2.2 Hardware/software mapping (mapping of sub-systems to processes and computers, communication between computers)

The mobile application runs on a mobile device, which has to use the Android operating system. The mobile application makes use of the Android SDK to connect to the database. The database also runs on the mobile device itself. The Android SDK uses the SQL language to communicate with the database.

2.3 Persistent data management (file/ database, database design)

The blockchain of a user is persisted in the SQLite database since we do not want to lose the data upon closing the application. The blockchain consists of blocks, and since the blockchain is persisted, the blocks are persisted too. Next, to that, The general information about the user, that we need for later, is saved in the database as well.

2.4 Concurrency (processes, shared resources, communication between processes, deadlocks prevention)

The users of the blockchain will have to communicate with each other, using a communication protocol, such as Bluetooth. Every user keeps their blockchain, and every other user can see this blockchain. The other user now can add the initial user to his own blockchain, and extend his chain. The users should also be aware of their own keys. In case these get stolen, the user should revoke it.

As for the shared resources, the idea is to keep everything decentralised. This means that there is no central server which keeps track of the blockchain. Every user keeps track of their own blockchain, and as stated earlier, the communication between users gives users the opportunity to know about the blockchain of others.

3 Glossary

- Android
  Android is an open-source operating system, which is developed by Google.
- Android SDK
  Android SDK is the software development kit, which is used for creating applications for the Android environment.
- Block
  A block contains information about another user. This information could be hashed.
- Blockchain
  A blockchain is a chain of blocks, of which a block contains the hash of the previous block.
- Bluetooth
  Bluetooth is an open standard for wireless connections between devices on a short distance.
- Cryptographic hashing function
  A cryptographic hash function is a hashing protocol, but it has some more properties, which ensure more security. For example, it is a one-way protocol; it is not feasible to find the input of the data using the result of the hashing function.
- Git
  Git is a way to control the versions of your program. You could update, delete or edit your version of the program using this.
- GitHub
  GitHub is a website that allows storage and parallel development of code for programming teams.
- Google
  Google is a large IT company which mainly focuses on developing software and hardware.
- Hash
  A Hash is a protocol to map any information to a particular arbitrary size.
- IBAN
  IBAN is an abbreviation for International Bank Account Number.
- Integration test
  An integration test tests whether multiple components can cooperate as expected.
- Java
  Java is a programming language for developing software.
- Middleware
  Middleware is the layer that enables a subsystem to communicate with another subsystem. One of these subsystems does not have to be in the application itself.
- Native application
  A native application is an application, which is built with the SDK of the device and therefore is operating system dependent.
- Open-source
  Open-source means that the source code of a program is available to use to the public.
- Public Key
  A public key is one of the two keys used in cryptography to encode information.
- Regression test
  A regression test tests whether new bugs have been introduced due to updated code.
- SHA-256
  SHA-256 is a cryptographic hashing function of the second SHA family.

- SQL
  SQL stands for Structured Query Language and is used to make mutations to a database.
- SQLite
  SQLite is a database engine which works with a SQL database.
- Test Driven Development
  Test Driven Development is a way of developing software, of which you write the tests first and build the program around it.
- Unit test
  A unit test tests a single component to verify its behaviour.
- Web application
  A web application is an application in the form of a website.
- Web-of-trust
  Web-of-trust is a concept in the cryptography to create the trustworthiness between a user and its public key.