Bitcoin payment framework on a social media platform

by

Wessel Wessels

Thesis presented in partial fulfilment of the requirements for the degree of Master of Engineering (Electronic) in the Faculty of Engineering at Stellenbosch University

Department of Electrical and Electronic Engineering,
University of Stellenbosch,
Private Bag X1, Matieland 7602, South Africa.

Supervisor: Prof. G. van Rooyen

December 2015

Declaration

By submitting this thesis electronically, I declare that the entirety of the work contained therein is my own, original work, that I am the sole author thereof (save to the extent explicitly otherwise stated), that reproduction and publication thereof by Stellenbosch University will not infringe any third party rights and that I have not previously in its entirety or in part submitted it for obtaining any qualification.

Date: 2015/12/10 ....

Abstract

Bitcoin payment framework on a social media platform

W. Wessels

Department of Electrical and Electronic Engineering,
University of Stellenbosch,
Private Bag X1, Matieland 7602, South Africa.

Thesis: MEng (E&E)

December 2015

English abstract to be written

Uittreksel

Bitcoin betalingsraamwerk op 'n sosiale media platform

("Bitcoin payment framework on a social media platform")

W. Wessels

Departement Elektries en Elektroniese Ingenieurswese,
Universiteit van Stellenbosch,
Privaatsak X1, Matieland 7602, Suid Afrika.

Tesis: MIng (E&E)

Desember 2015

Afrikaanse uittreksel wat nog geskryf moet word

Acknowledgements

I would like to express my sincere gratitude to the following people and organisations ...

Dedications

Hierdie tesis word opgedra aan ...

Contents

List of Figures

List of Tables

Nomenclature

No nomenclature yet.

Chapter 1

Introduction

Bitcoin [1] is a peer-to-peer decentralised payment mechanism

1.1 Background

1.2 Related Work

1.3 Objectives

Chapter 2

Background

2.1 History

Chapter 3

System Design

3.1 Framework

In this project we test the viability of a payment framework on a mobile social media platform. The framework will consist of several independant but connected pieces:

- REST API for payments
- Wallet Application
- Bitcoin Interface
- Use-case Application

A summary of what is required from the system can be seen in figure 3.1.

3.1.1 REST API for payment management

A REST (Representational State Transfer) API [2] was chosen for the main interface for developers to use Bitcoin without running a Bitcoin node or having experience with Bitcoin. REST was chosen because it is a commonly used architecture, it is easy to use and understand and it does not constrain the user's choice of programming langauge or environment.

The purpose of the REST API is to let developers make payment requests and check if a payment has been made, without dealing with the low-level Bitcoin transactions directly. Thus, our system should generate a new Bitcoin address on request.

Our requirements from the REST API are:

- New Bitcoin address for each payment
- Verify payment

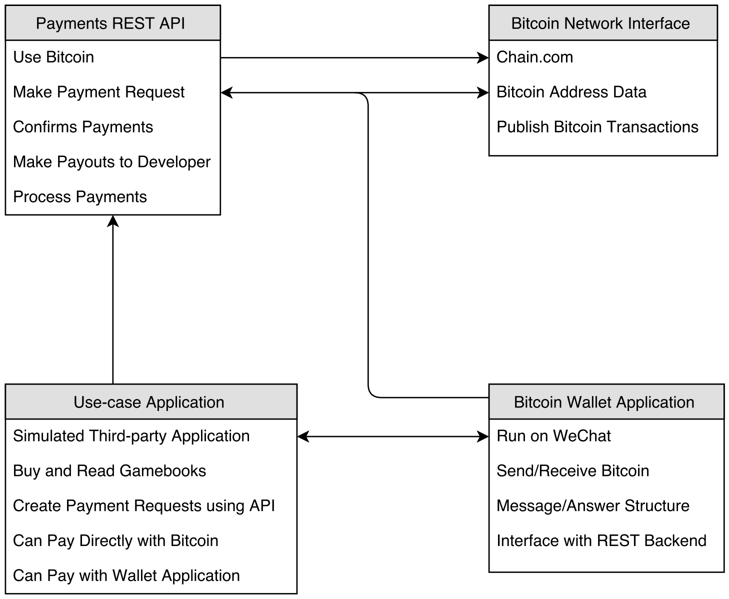
Payments REST API
Use Bitcoin
Make Payment Request
Confirms Payments
Make Payouts to Developer
Process Payments
Bitcoin Network Interface
Chain.com
Bitcoin Address Data
Publish Bitcoin Transactions
Use-case Application
Simulated Third-party Application
Buy and Read Gamebooks
Create Payment Requests using API
Can Pay Directly with Bitcoin
Can Pay with Wallet Application
Bitcoin Wallet Application
Run on WeChat
Send/Receive Bitcoin
Message/Answer Structure
Interface with REST Backend

Figure 3.1: Summary of Framework

- Check total balance of developer
- Witdraw available Bitcoin of developer

3.1.1.1 The concept of the Bitcoin payment

This is a high-level explanation of how to receive verifyable payments with Bitcoin. With Bitcoin, unlike a traditional bank account, you don't have a single "account" where people can make payments to and you can verify that the payment came from them. With Bitcoin it is trivially easy to make a new Bitcoin address, and it can be generated without being connected to the Internet or the Bitcoin network.

Since the entire Bitcoin blockchain is public, a single address is not sufficient to receive multiple payments. With a single address, it is not easy to verify that a spesific person has made a payment, since there may be several payments of the same amount happening in short succession.

The sollution to the problem is generating a new address for every payment, and requesting that the user make the payment to that address. Since the newly generated address is not yet present on the blockchain, when a payment to that address of the requested amount occurs, we can be certain that the person in question made the payment. When the payment is complete, the Bitcoin in that address can be transferred to a central address, and the original address can be discarded.

From our requirements for the REST API, we clearly require (at least) the following methods:

- A “payment” method
- A “balance” method
- A “payout” method

3.1.1.2 The /payment method

The /payment method is the core of the REST API. It is used to make a payment request with a specified amount of Bitcoin and a description of the transaction. The /payment method returns a Bitcoin public address and a payment ID.

The user can then pay to the Bitcoin address using any standard Bitcoin payment method, or can pay directly from the Bitcoin wallet that will run on WeChat and will be connected to the payment infrastructure.

3.1.1.3 /payment/{PUBLIC_ADDRESS} and /payment/{ID}

These two methods are conceptually the same, but they take in two different arguments. The one takes the Bitcoin address to be queried, and the other takes the payment ID. The method returns all the data about the transaction, including the status of the transaction.

The main purpose of this method is to verify that a transaction has been completed by the user. It can also be used to give the payment details to to user again.

3.1.1.4 /balance

The /balance method gives the developer the balance of all the available Bitcoin from all the received transactions. The method also returns a flag that says if there is enough Bitcoin to make a payout.

3.1.1.5 /payout

The /payout method is used by the developer to transfer all of the available Bitcoin to a specified Bitcoin address.

3.1.2 Wallet Application

The Wallet Application is a Bitcoin wallet implemented on the WeChat platform. The WeChat platform uses a simple message-answer structure. A user sends a message in the Wallet Application. The message is then sent to WeChat that sends it to a third party server controlled by the developer. The server then sends a reply to WeChat that is then forwarded to the user.

In this manner, a fully functional Bitcoin wallet is realised. The third party server stores the private keys and processes the Bitcoin transactions on commands from the user.

The advantage of using the WeChat platform is the security built in to the platform, as well as an existing userbase.

3.1.3 Bitcoin Interface

To connect to the Bitcoin peer-to-peer network, a Bitcoin client is needed. The standard way of doing this is running the Bitcoin open source software on a server. This is very network and processor intensive. Thus, an alternative is used for interfacing with the Bitcoin network.

We require the following from such an interface:

- Get the balance from an address,
- Get unspent outputs from an address,
- Post a signed Bitcoin transaction to the network,
- It must be able to use the Testnet

After considering several options, a service called chain.com was chosen. Chain.com is a free service that satisfies all the requirements. It is perfect to use this service as a proof of concept, but in practice one would rather run a full Bitcoin node to minimize reliance on third-party services.

3.1.4 Use-case Application

To use the payment framework, a Gamebook application is created to read Choose Your Own Adventure style books on the WeChat platform. User-created books can be sold by using the Bitcoin payment framework and the author can potentially earn Bitcoin.

For each sale, the Gamebook application creates a payment request using the REST API. The user can then pay using the Wallet Application or any Bitcoin payment mechanism. The Gamebook application can the query the API to confirm that the payment is received.

Chapter 4

Detail Design

4.1 Back-end

To implement the payment framework, we need a back-end server and a back-end web framework.

4.1.1 Back-end Server

We chose an Amazon Elastic Cloud Computing (EC2) instance for the back-end server. EC2 gives us access to a virtual machine (VM) where we can run our own software, including a publicly accessible website. The micro instance is deployed in Singapore in the Asia Pacific region. The reason for this is to have the server as close as possible to the WeChat servers in China, since our servers must connect to WeChat's servers.

4.1.2 Back-end Web Framework

We decided to use XAMPP (Apache, MySQL, PHP and Perl) for the back-end development environment. XAMPP is a full stack development environment. It includes an HTTP server (Apache), a database server (MySQL) and a scripting language (PHP). XAMPP is free and easy to deploy, and is also used because the author is familiar with it.

4.2 Social Media Platform

We chose WeChat for our social media platform. WeChat works on most smartphones, already has a userbase and it has a third-party API with a development sandbox feature.

On WeChat, a third-party application is known as an "Official Account". We registered for a sandbox Official Account that only allows 20 users and is not searchable on WeChat.

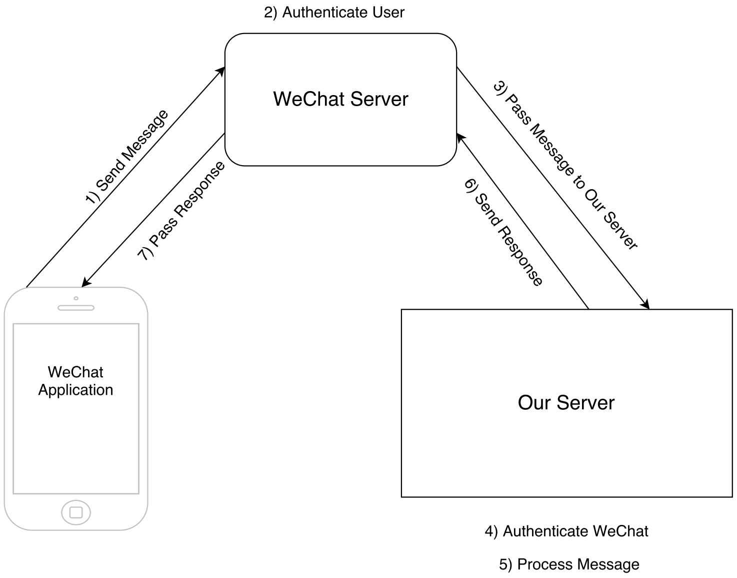

Figure 4.1: Interaction with WeChat

WeChat acts as an intermediary between the user and our server as seen in figure 4.1.

WeChat uses a shared token hashing scheme to authenticate itself on our service. We provide

Chapter 5

Tests

5.1 Quantative

5.2 Qualitive

Chapter 6

Conclusion

6.1 The Conclusion

Appendices

Appendix A

No appendices yet

Bibliography

[1] S. Nakamoto, “Bitcoin: A Peer-to-Peer Electronic Cash System,” *Consulted*, pp. 1–9, 2008. [Online]. Available: https://bitcoin.org/bitcoin.pdf

[2] Oracle.com, “What Are RESTful Web Services?” [Online]. Available: https://docs.oracle.com/javaee/6/tutorial/doc/gijqy.html