The Report Committee for David Allen Bronleewe
Certifies that this is the approved version of the following report:

Bitcoin NFC

APPROVED BY
SUPERVISING COMMITTEE:

Supervisor: _____
Christine Julien

_____
Bill Bard

Bitcoin NFC

by

David Allen Bronleewe, B.S.

Report

Presented to the Faculty of the Graduate School of

The University of Texas at Austin

in Partial Fulfillment

of the Requirements

for the Degree of

Master of Science in Engineering

The University of Texas at Austin

August 2011

Abstract

Bitcoin NFC

David Allen Bronleewe, M.S.E.
The University of Texas at Austin, 2011

Supervisor: Christine Julien and Bill Bard

Bitcoin is a new digital, virtual currency. Unlike other currencies, it is not controlled by a single company, financial institution or government. Instead, it is controlled by a peer-to-peer network of clients running the open-source Bitcoin software.

When using Bitcoin, online transactions can be made directly between two parties without the need for a trusted third party such as Paypal, Visa or a bank.

This paper describes the development of Bitcoin NFC, an Android app designed to send bitcoins from one Android device to another.

The technology used for sending information between devices is Near Field Communication (NFC), a very close-ranged wireless protocol. NFC allows devices to communicate by simply touching two devices together. There is no need for any configuration or pairing.

Bitcoin NFC makes it possible to use bitcoins for point-of-sale transactions. Rather than swiping a credit card, a phone running Bitcoin NFC could be swiped instead.

Table of Contents

List of Figures

1. Introduction

Smart phones, such as those running Android or iOS, are becoming more and more capable with each new generation of devices. They can serve many different functions such as a camera, video camera, GPS unit, web browsing device communications device of all sorts: voice, video, text message, instant message, e-mail.

Google now wants people to also use their phones for mobile payments. Rather than swiping your credit card, the phone would be swiped instead. Google's new mobile payment service is called Google Wallet. (1)

The technology used to make this swiping possible is called Near Field Communication (NFC), a very close-ranged wireless protocol.

I developed Bitcoin NFC, an Android app, to enable mobile payments between Android devices. Rather than transferring US dollars like Google Wallet, Bitcoin NFC transfers a new virtual currency called Bitcoin.

At the time development began on this project, there were no other mobile clients for Bitcoin. (2) Currently there is a more fully featured app called Bitcoin Wallet that can be found on the Android Market. (3)

1.1 About Bitcoin

Bitcoin is a new digital, virtual currency. Unlike other currencies, it is not controlled by a single company, financial institution or government. Instead, it is controlled by a peer-to-peer network of clients running the open-source Bitcoin software. (4)

There are a few organizations accepting donations in bitcoins and several small businesses accept them, but Bitcoin has yet to see widespread adoption. It will be interesting to see if Bitcoin usage will ever become ubiquitous.

1.1.1 Economics

Bitcoins can be freely exchanged between parties. They can be used to pay for goods or services, and there are exchange markets where bitcoins can be bought or sold using other currencies. They are not backed by gold or any other currency. Their value is based on scarcity. The Bitcoin system is designed so that there is always a limited number of bitcoins in circulation.

The value of bitcoins on the exchange markets have increased drastically. At the beginning of 2011, a single bitcoin sold for under $1 USD. By June, they went up as high as $30 USD. As of August, 2011, the price has leveled off at around $10 USD per bitcoin.

The most popular exchange is Mt. Gox. (5) Once an account is created with Mt. Gox, you can use funds from a bank account in order to purchase bitcoins from other Mt. Gox users.

1.1.2 Transactions

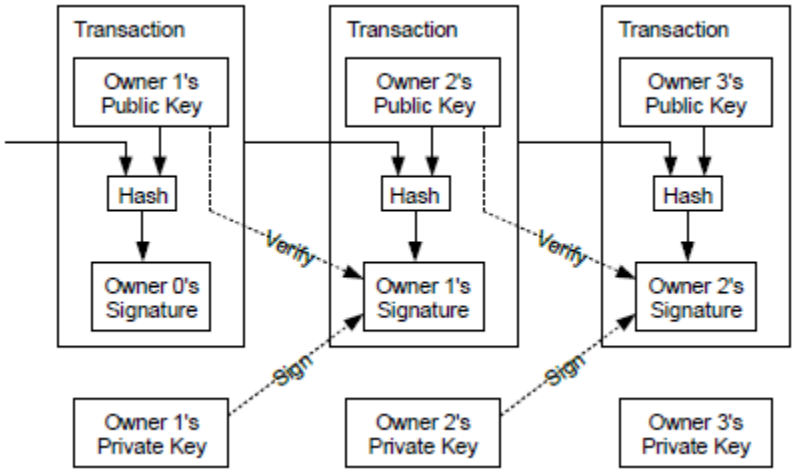

Figure 1: Bitcoin Transactions (4)

Transactions occur within Bitcoin using public key cryptography. For example, Alice wants to send Bob a bitcoin. The bitcoin's previous transaction is hashed along with Bob's public key. Alice then signs this hash with her private key. A transaction is created that consists of Bob's public key, the hash and the signature. Anybody can use Alice's public key to verify the signature and prove that Bob now owns this bitcoin. Bob is now the only one who can spend this bitcoin since it requires his private key to do so.

When a transaction is created by a Bitcoin client, it is sent to all the peers that are connected to the client. These peers then forward the transaction to their peers and the transaction propagates through the Bitcoin network.

The Bitcoin protocol uses elliptic curve cryptography (ECC) for its public/private key operations. The RSA algorithm has been around longer, so it is more commonly used. But ECC can provide the same amount of protection while using smaller keys and executing more efficiently. (6)

A derivation of the public key is used as a Bitcoin address. Each client can have any number of addresses. It stores its public and private keys in a file. The main Bitcoin client calls this file *wallet.dat*. Anybody with possession of the *wallet.dat* file has the ability to spend the bitcoins associated with the addresses it contains.

A Bitcoin transaction can contain several inputs and several outputs. Each of the hashes of the input transactions must be signed by their corresponding private keys. The values of the bitcoins from the inputs are summed and can be split between multiple outputs if desired.

Transactions can contain fractional parts of bitcoins. The smallest amount supported by the Bitcoin protocol is 0.0000001.

1.1.3 Block Chain

Creating transactions using public key cryptography prevents forgery. But it does not prevent the owner of the bitcoin from spending it more than once. This is called double spending.

To prevent double spending, the first transaction that occurs needs to be regarded as the real transaction and the other discarded. On a single server solution, it is a simple matter to determine which is first. But with a peer-to-peer network, a more elaborate solution is required.

Transactions are grouped together into blocks as seen in Figure 2. Each block includes a hash of the previous block. This forms a sequence of blocks called the block chain. The block chain is distributed to all clients. The first of the duplicate transactions to make it into the main block chain will be considered to have succeeded. The others will be discarded. So even after receiving a signed transaction, the recipient must wait until that transaction has made it into the block chain before they can trust that they are the true owner of the bitcoins. (7)

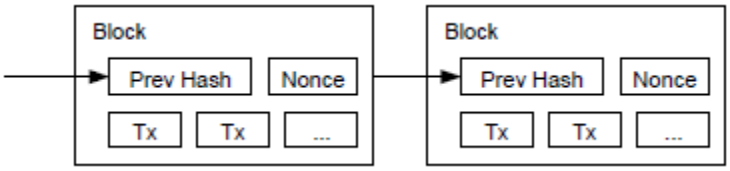

Figure 2: Block Chain (4)

When new blocks are discovered they are sent out to the Bitcoin client's peers, which in turn forward the new block to its peers and so on until the entire peer-to-peer network has a copy of the latest block.

1.1.4 Bitcoin Mining

Blocks are designed to be computationally difficult to create. In order for a block to be valid, its hash must start with a predetermined number of zeros. Since

hashes are one way functions, the only way to do this is by brute force. Along with the transactions and the hash of the previous block, each block includes a nonce, some added bits. The nonce bits are set to random numbers until a hash of the block is found that begins with a required number of zeros.

This computational complexity of creating blocks is what prevents malicious clients from overtaking the block chain with is its own set of blocks that could be used to rewrite the transaction history.

The difficulty of finding blocks scales with the computational power of the network. The Bitcoin protocol is designed so that new blocks should be discovered every 10 minutes on average. To achieve this, the hashing difficulty is scaled by changing the number of bits in the hash that are required to be zeros. With each additional required bit, the difficulty doubles.

Right now, the first transaction included in a block is always for 50 bitcoins. It is assigned to the address of the miner who discovered it. So there is financial incentive for clients to contribute to the strength of the network.

As long as no single malicious client or group of clients makes up over 50% of the network, the good clients will create blocks at a faster rate. So, if by chance, malicious clients create several blocks in a row, it is just a matter of time before the rest of the network will overtake the forked chain.

1.1.5 Development

The Bitcoin protocol was designed by Satoshi Nakamoto, who wrote a paper in 2009 describing the core required principles. (4) It is unsure whether or not this is his real name or just a pseudonym. (8)

The code for the primary Bitcoin client is set up as an open-source project. The original code was written by Nakamoto and has been maintained and updated by Andreas Gavin and others. This client is written in C++. (9)

A Java implementation called BitCoinJ was written by a Google employee during his 20% time (time where Google employees can work on a project of their choice). BitCoinJ is also open-source and others are now helping to contribute to the project. (10)

1.2 About Near Field Communication

Near Field Communication, also referred to as NFC, is a short-ranged wireless protocol. Small amounts of data can be transferred between devices that are in contact with each other, or at least within centimeters of each other. Only one of the devices is required to be powered for most NFC operations. The unpowered device will get power from the signal of the powered device. This allows for very small target devices. These passive devices are called NFC tags. Android phones are able to present themselves to other NFC devices as though they were NFC tags.

NFC operates at the 13.56 MHz frequency and has a range of around 4 cm. The data rate can be 106, 212 or 424 kbps when one device is passive. When both are active, the rate goes up to 6,780 kbps. (11)

The NFC Forum is the organization in charge of certifying devices and maintaining the specification. The organization was formed in 2004 by Nokia, Philips and Sony.

1.3 About Android

Android is an open-source, mobile OS developed by Google. Currently, Android has more of the smartphone market share than any other mobile OS.

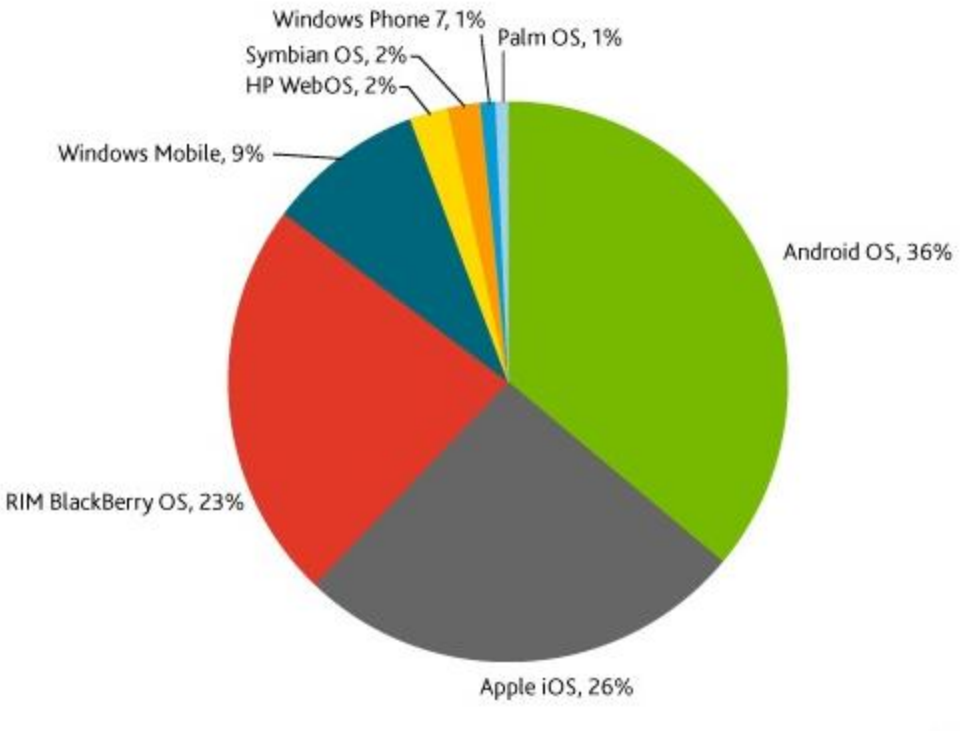

Figure 3: Smartphone Market Share - April 2011 (12)

The core OS itself is based on Linux. A Java layer was created on top of that. Apps for Android are written in Java using the Android SDK, which contains the Java libraries needed for apps to interact with the OS.

The latest version of Android for phones is 2.3, also named Gingerbread. The minor update 2.3.4 introduced the API for using Near Field Communication.

2. Development

This section describes the development effort that went into creating Bitcoin NFC. Source code can be found at https://code.google.com/p/bitcoin-nfc/. The project is open-source under the Apache license.

2.1 HARDWARE

The following is a list of hardware used in the development Bitcoin NFC:

- Dell Inspiron 1720
  - 2.4 GHz Intel Core 2 Duo
  - 4 GB RAM
  - Running Windows 7 (x64)
- Samsung Nexus S
  - Running Android 2.3.4
- Samsung Nexus S
  - Running Android 2.3.4

2.2 SOFTWARE

The following is a list of software used in the development Bitcoin NFC:

- Android SDK Tools, revision 12
- Android SDK Platform-tools, revision 6
- SDK Platform Android 2.3.3, API10, revision 2
- Java SE Development Kit 6 Update 26 (x64)
- IntelliJ IDEA Community Edition 10.5.1
- Bitcoin 0.24
- Mercurial 1.9.1 (x64)
- TortoiseSVN 1.6.16 (x64)

2.3 ARCHITECTURE

Figure 4 lists the Java files created for Bitcoin NFC. Figure 5 shows an overview of how the Bitcoin NFC code implements the BitCoinJ library and

Android's NFC code. For both BitCoinJ and NFC, a single file was used to interface with these external libraries. This way any changes to the external library would only need to be updated in a single file rather than all throughout the code.

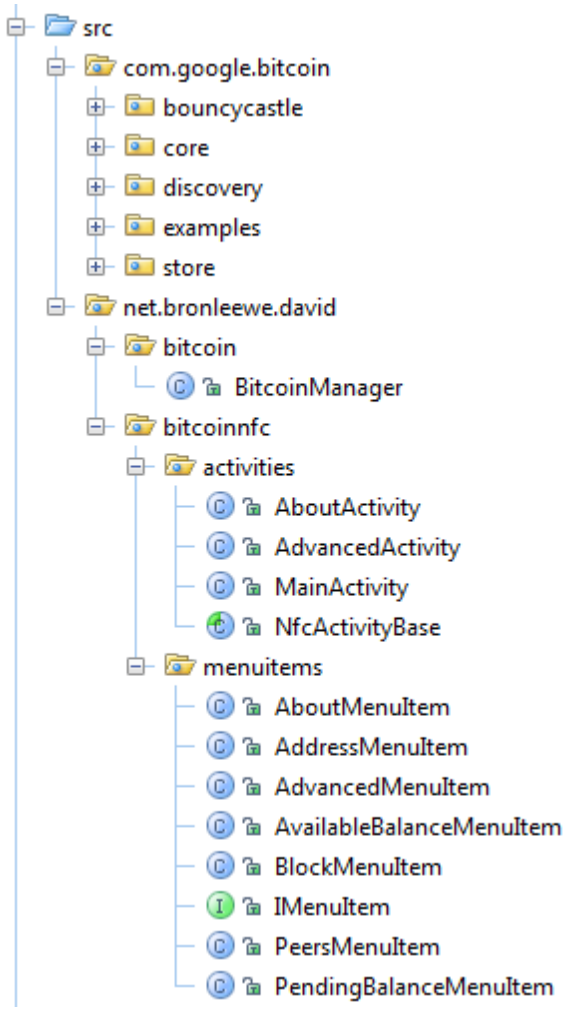

Figure 4: Java files created for Bitcoin NFC

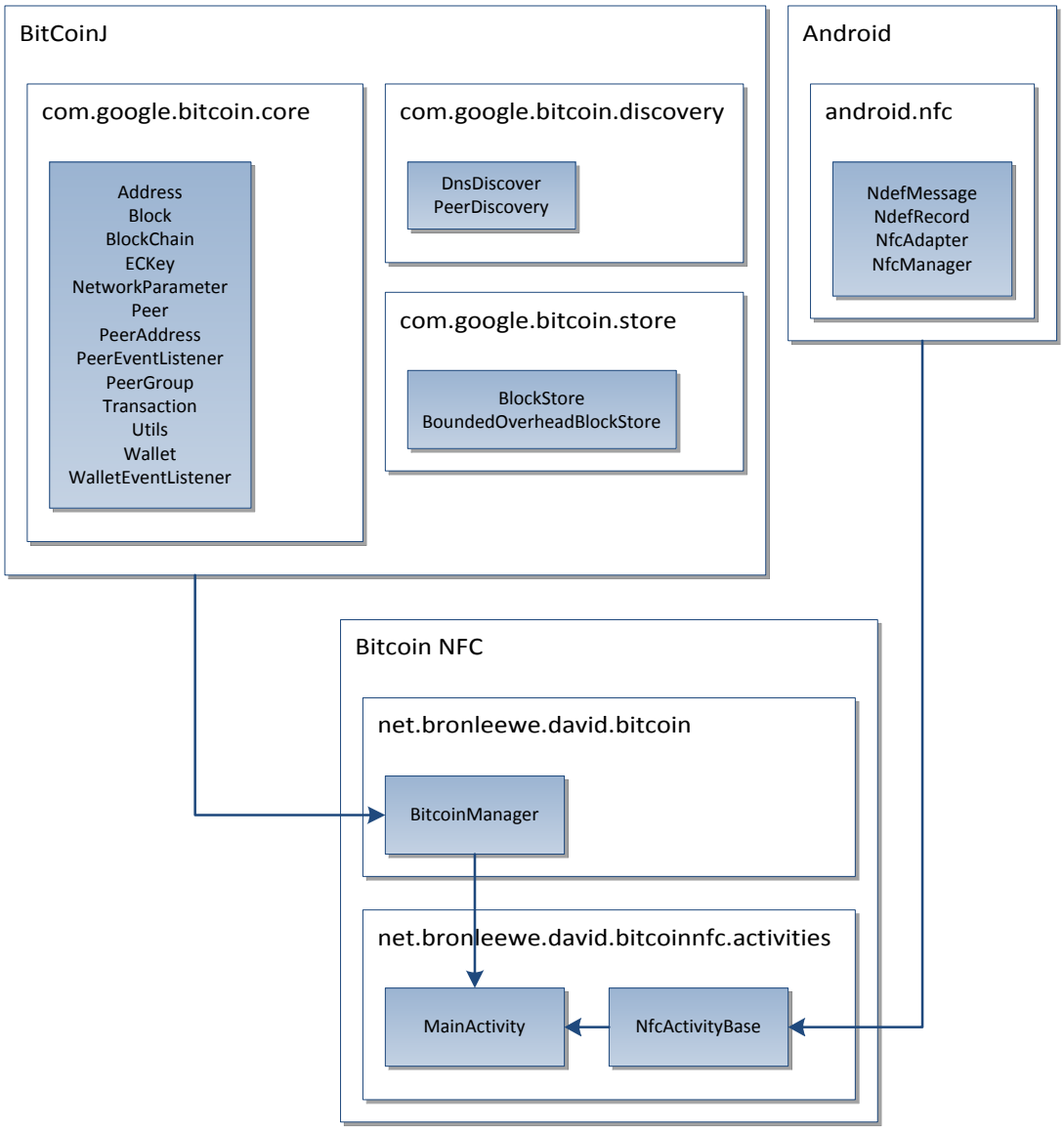

Figure 5: Bitcoin NFC Architecture

2.4 USING BITCOINJ

BitCoinJ is a Java implementation of the Bitcoin client code. The original Bitcoin client was written in C++ and is not as easy to work with. BitCoinJ is carefully architected as reusable Bitcoin libraries that other developers can implement when creating their own Bitcoin client software or apps. It was

developed by a Google employee and released as open-source under the Apache license. (10)

Using the BitCoinJ library for Bitcoin NFC app was very straightforward. The code is well documented and comes with many samples. Since it is written natively in Java, there were no issues getting it to compile and run on Android.

It is still being actively developed. Throughout development, BitCoinJ was regularly updated to get the latest features and bug fixes.

One new feature recently added to BitCoinJ is PeerGroup. Normally a Bitcoin client will attempt to connect to several peers. Prior to this new feature, each peer, represented by the Peer object had to be individually managed. Now, all the peers can be added to a PeerGroup. The PeerGroup connects and disconnects peers, downloads the block chain and dispatches messages. For instance PeerGroup.sendMessage() will automatically send a message to all currently connected peers. The addition of PeerGroups simplified the Bitcoin NFC code and made the BitCoinJ code much easier to work with.

2.5 USING NEAR FIELD COMMUNICATION

For the initial implementations of NFC, Bitcoin NFC was able to receive any type of NFC message. Google provides an app called Tags where virtual tags can be created on the phone. I was able to create a tag on one phone and when placed up against the other phone, Bitcoin NFC would launch. The next step was to parse the NFC message and then display the result in a TextView. After watching Google's NFC presentation from Google IO 2011, I learned the correct way for my app to send and receive app-specific NFC messages.

In the AndroidManifest.xml instead of specifying text/plain for the MIME type, I filtered on application/vnd.bitcoin so that only these types of tags will be handled by Bitcoin NFC.

2.6 USING ACTIVITIES

The Android API divides up pieces of an application into "activities." Each activity can define its own UI and execute code. When an activity is created, it is pushed to the top of the stack of activities. When the user presses "back," it is then removed from the stack.

An example of this would be the Gmail app. Opening the app pushes the Inbox activity to the front. Tapping a message pushes that e-mail activity to the front. Pressing "back" once pops you back to the Inbox. Pressing "back" once more pops the Inbox activity which closes the app.

Bitcoin NFC uses three activities: MainActivity, AdvancedActivity and AboutActivity. AdvancedActivity and AboutActivity are both pushed when their corresponding list entries are pressed. Pressing "back" from those screens then returns you back to MainActivity. Another "back" press will pop the final activity, thus exiting the program.

Each activity can respond to several events to help them get started or clean things up before exiting. The lifecycle of an activity along with the overloadable methods is shown in Figure 6.

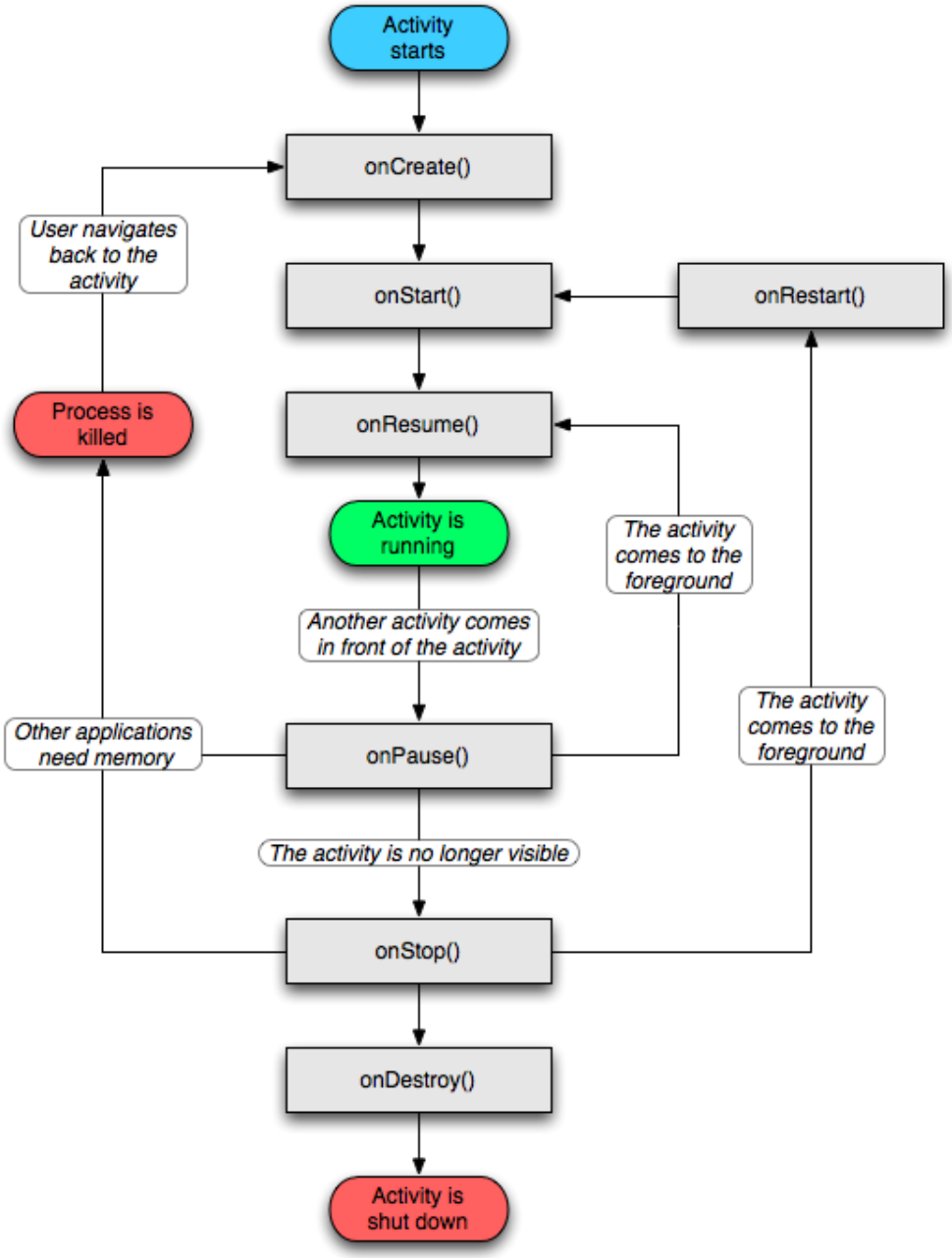

Figure 6: The Activity Lifecycle (13)

3. Bitcoin NFC

This section describes the usage model for Bitcoin NFC and explains the UI.

3.1 USAGE OVERVIEW

An overview of the sequence of events for using Bitcoin NFC can be found in Figure 7. The steps for sending bitcoins from one device to another are as follows:

1. The device wishing to receive bitcoins launches Bitcoin NFC.
2. The two devices are tapped together.
3. Bitcoin NFC automatically launches on the sending device with the receiving device's Bitcoin address already filled in.
4. The sender enters the amount they wish to send and tap the Send button.
5. A Bitcoin transaction is created and sent out to the network.
6. After the transaction has made it into a block, the new balance will be updated on the receiving device.

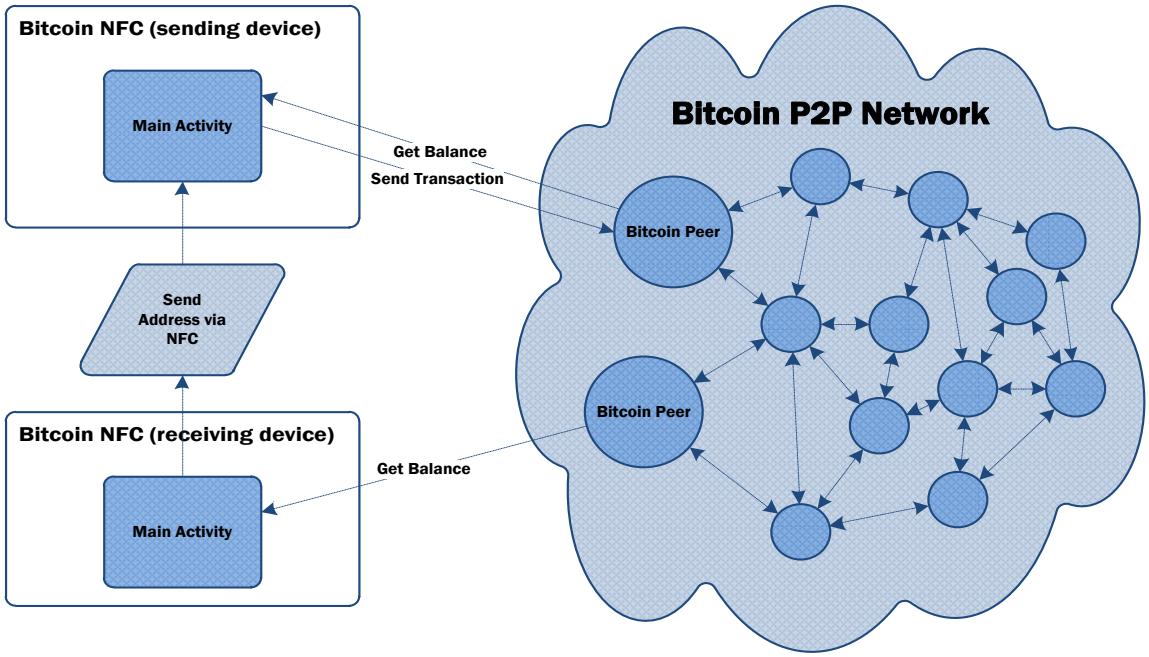

Figure 7: Bitcoin NFC Operation

3.2 MAIN SCREEN

The bulk of the work is done in the MainActiviy. It starts up the Bitcoin client during onCreate() and creates an NFC readable message in onResume(). The message will send the Bitcoin address that is currently loaded.

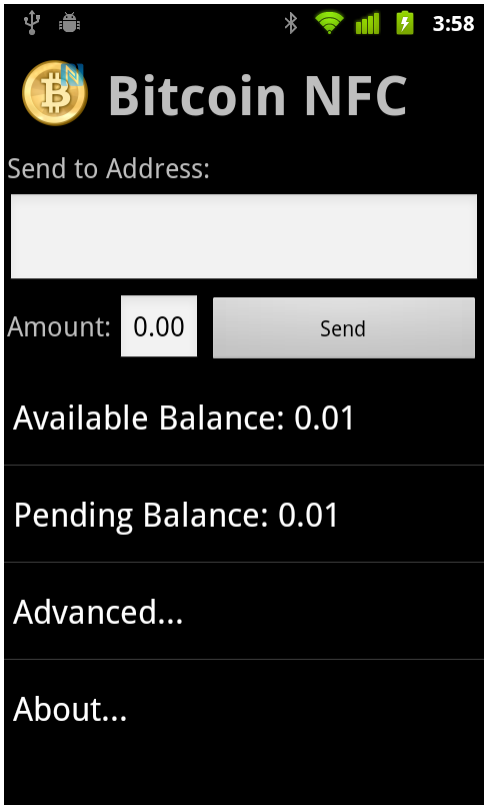

Figure 8: Main Screen

3.2.1 Send to Address

This text box contains the Bitcoin address where the bitcoins are to be sent. It can be auto-populated by using NFC. An address obtained by some other means can also be pasted into the box.

3.2.2 Amount

The user types in the number of bitcoins they intent to send.

3.2.3 Send Button

Tapping the Send button will create a transaction with the specified number of bitcoins being sent to the specified address. It will fail to send if this amount is greater than the available balance.

3.2.4 Available Balance

The Available Balance item displays the current number of bitcoins contained within the wallet that are available to spend. Transactions that are not yet confirmed will not be included in this balance.

3.2.5 Pending Balance

The Pending Balance item will take into account transactions that have been received, but have not yet been included in enough blocks to be fully verified.

3.2.6 Advanced…

This item launches the Advanced Screen.

3.2.7 About…

This item launches the About Screen.

3.3 ADVANCED SCREEN

The internal Bitcoin information is located on the Advanced screen.

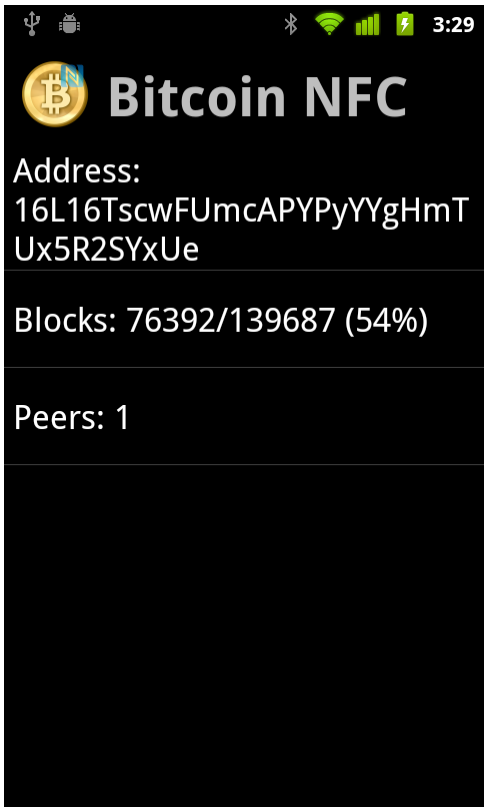

Figure 9: Advanced Screen

3.3.1 Address

The Address item shows the Bitcoin address that has been saved to your wallet file. Others can send bitcoins to this address and those transactions will be saved to your wallet when the client receives the block containing the transaction.

Pressing the Address item in the list will copy your address to the clipboard. This allows the address to be sent to somebody using e-mail, text message, instant message or any other method desired.

3.3.2 Blocks

The Blocks item shows the number of blocks downloaded over the number of blocks needed since the app was last launched. The first time the app is launched, it can take some time to download the entire block chain from its connected peer.

Subsequent launches of the app will be much quicker since only the blocks that were generated since the app was last running will need to be downloaded.

For debugging purposes, pressing the Blocks item will delete the block chain file so that the next time the app is launched, it will re-download the entire block chain.

3.3.3 Peers

The Peer item shows the current number of bitcoin clients that that are connected to Bitcoin NF. For testing purposes, I used testnet, a parallel Bitcoin system used only for development, rather than the production Bitcoin network. Bitcoin NFC connects to my local machine running the standard C++ Bitcoin client in testnet mode seen in Figure 10.

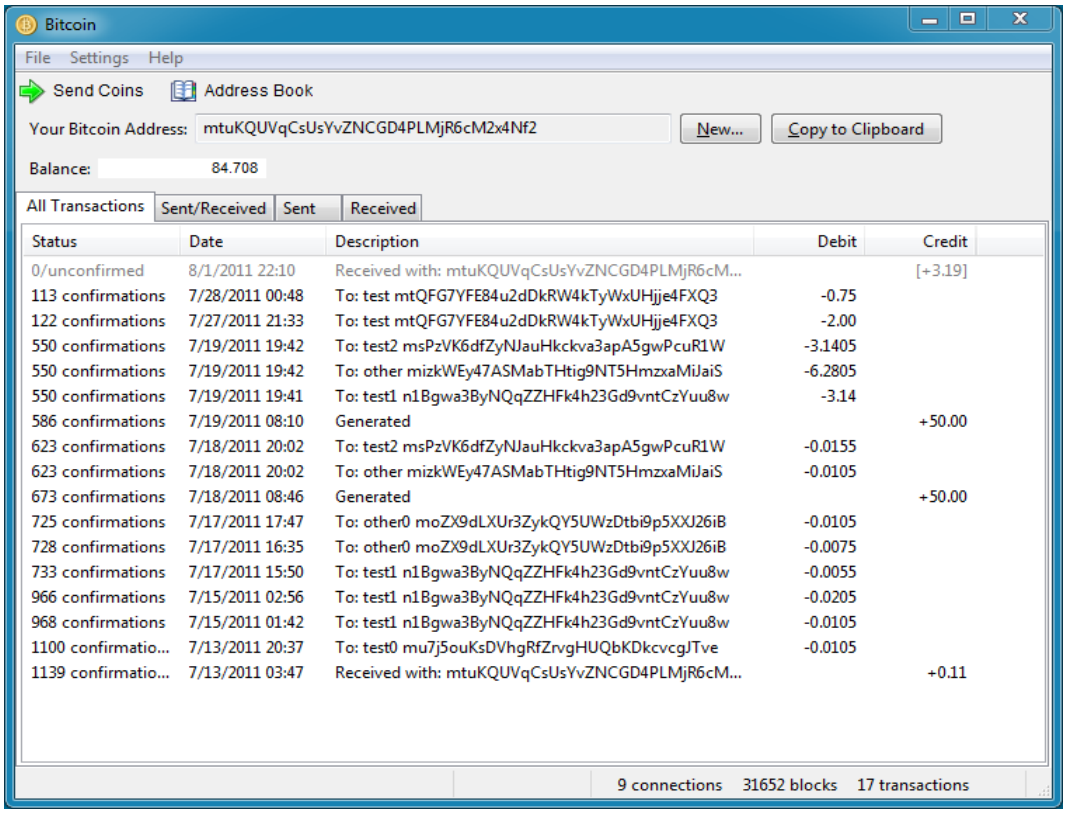

Figure 10: PC Bitcoin client, connected to phone app

3.4 ABOUT SCREEN

The About screen displays the version number and information about the app's author. Pressing "back" returns the user to the main screen.

Figure 11: About Screen

4. Future Work

4.1 Sender Initiated NFC

My original goal was to have the sender be the NFC initiator. The steps for this protocol would be as follows:

1. The sending user inputs the number of bitcoins to send and presses the send button.
2. The phones are tapped together.
    a. The sending device transmits the number of bitcoins over NFC.
    b. The receiving device immediately transmits back its bitcoin address.
3. The sending device uses this address to create a transaction and send it out onto the Bitcoin peer-to-peer network.

I was unable to find a way to implement this protocol using a single tap. After receiving an NFC message, I had the app immediately respond with another NFC message. This caused an exception in the app and it would not work. Currently, Android seems to only support sending one message per tap.

In order for this approach to work, it would require multiple taps of the devices. I opted, instead, for a simpler approach where the receiving device initiates the transaction by sending its address.

Based on the Android documentation, it is unclear if sending multiple NFC messages during a single tap is something that should work. Perhaps it will be supported in a future API. It would make the app more intuitive by letting the sender initiate a transaction.

4.2 Offline Senders

The NFC exchanges could be set up in such a way that only one of the devices needs to be connected as a peer to the Bitcoin network. This could be useful in a point-of-sale transaction where the payment device is connected to the Internet, but the phone is not required to have a data connection.

4.3 FULLY USABLE CLIENT

There are many features that need to be added to Bitcoin NFC to make it a more useful Bitcoin client:

- A means to encrypt and backup the wallet file.
- Address book for storing Bitcoin addresses that are frequently used.
- A method for requesting a specified amount of Bitcoins.
- A way to switch between using testnet and the production network.

4.4 PASSIVE NFC

Passive NFC tags can be made inexpensively and in a variety of small form factors such as credit card sized cards or even stickers. (14) Passive NFC devices are called tags. NFC capable phones can read the data on these tags and write to them also.

Assuming that the use of Bitcoin has become ubiquitous, the following is a potential application for passive devices.

A family spends the day at the fair. The children want to go on some rides and buy some snacks. The father hands them each a card or token that, on the spot, using his phone, he is able to program with a certain amount of bitcoins. Now if the card or token is lost or stolen, the impact is minimal. The alternative would be that each child would be carrying around a much more expensive electronic device.

To use NFC tags in this way, bitcoins are sent to a temporary address generated by the phone or other NFC capable device. The public/private key pair is written to the NFC tag. The possessor of the card or token is able to read the key pair and claim the bitcoins. (15)

There are limitations to this method. A person writing bitcoins to the tag will also have the private key needed to reclaim the bitcoins. The card or token itself should not be accepted from a non-trusted person and used as payment. The bitcoin data needs to always be read off the card or token and transferred to the recipient's own address in order to be used as payment.

5. Related Works

5.1 Bitcoin Wallet

Bitcoin Wallet is an Android app, created by Andreas Schildbach. The app is capable of sending bitcoins between Android devices using QR codes. It has an address book to store frequently used addresses. This project is open-sourced under the GPL and, like Bitcoin NFC, uses the BitCoinJ library. (3)

Figure 12: Bitcoin Wallet

As of version 1.10, released July 16th, Bitcoin Wallet added NFC support. The method used is the same as that used by Bitcoin NFC, where the recipient sends an NFC message which contains the recipient's address. In addition to this, Bitcoin Wallet lets the recipient request an amount of bitcoins.

5.2 GOOGLE WALLET

Google Wallet is a point-of-sale NFC payment service that will allow you to make payments using your NFC enabled Android device. It is currently only rolled out at select locations and only works with the Sprint variant of the Nexus S. The T-Mobile and AT&T variants of the phone are not yet supported. Google's plan is to expand the number of locations and supported devices over time. (1)

6. Conclusion

Bitcoin is an exciting new technology with a lot of potential new applications. The idea that anybody can create their own program that interacts with this currency is a game changer. There is no approval process, no authorization required. Bitcoin is open for development.

NFC is the perfect technology for securely passing information from one device to another. There is no configuration required. For Bitcoin NFC, when the main screen is displayed on one device, the other device merely needs to be placed up against it and the address is filled in and bitcoins are ready to be sent.

Currently NFC is not in widespread use. If it becomes available in more devices, it has the potential to open up new ways for our devices to interact. This includes secure payments and other financial transactions. It has the advantages of being secure, since it requires such close proximity, and no configuration or pairing is required for the devices to begin communication.

Developing for Android takes some time to learn, but there is a lot of documentation and sample code available on the Android development site. With the huge number of Android devices in use, this is a very useful platform to learn to program for. Also, unlike iOS and Windows Phone, there is no approval process required for an app to be released to others. Anybody can download the SDK for free and develop something useful for others.

I was able to accomplish my main objective. I created Bitcoin NFC, which is capable of sending bitcoins from one device to another. The code can be found at http://code.google.com/p/bitcoin-nfc/. Others can use this code for their own apps or projects.

Bibliography

1. FAQ. *Google Wallet.* [Online] http://www.google.com/wallet/faq.html.

2. Android Bitcoin Client Bounty. *Bitcoin Forum.* [Online] https://bitcointalk.org/index.php?topic=1812.0.

3. Bitcoin Wallet. *Android Market.* [Online] https://market.android.com/details?id=de.schildbach.wallet.

4. **Nakamoto, Satoshi.** *Bitcoin: A Peer-to-Peer Electronic Cash System.* 2009.

5. *Mt. Gox.* [Online] https://mtgox.com/.

6. The Case for Elliptic Curve Cryptography. *National Security Agency.* [Online] January 15, 2009. http://www.nsa.gov/business/programs/elliptic_curve.shtml.

7. Block chain. *Bitcoin Wiki.* [Online] https://en.bitcoin.it/wiki/Block_chain.

8. Satoshi Nakamoto. *Bitcoin Wiki.* [Online] https://en.bitcoin.it/wiki/Satoshi_Nakamoto.

9. bitcoin. *github.* [Online] https://github.com/bitcoin/bitcoin.

10. BitCoinJ. *Google Code.* [Online] http://code.google.com/p/bitcoinj/.

11. *Information technology — Telecommunications and information exchange between systems — Near Field Communication — Interface and Protocol (NFCIP-1).* s.l. : ISO/IEC 18092, 2004.

12. NIELSEN: Android Still Clobbering Apple's iPhone, RIM's BlackBerry. *Business Insider.* [Online] http://www.businessinsider.com/nielsen-android-still-clobbering-apples-iphone-rims-blackberry-2011-5.

13. Activities. *Android Developers.* [Online] http://developer.android.com/guide/topics/fundamentals/activities.html.

14. *tagstand.* [Online] http://www.tagstand.com/.

15. Send bitcoins to an unknown recipient. *Bitcoin Forum.* [Online] http://bitcointalk.org/?topic=3427.0.

16. **Haselsteiner, Ernst and Breitfuß, Klemens.** *Security in Near Field Communication (NFC).* 2006.

17. **Grinberg, Reuben.** *Bitcoin: An Innovative Alternative Digital Currency.* 2011.

18. **Gibson, Steve.** Security Now 287: BitCoin CryptoCurrency. *This Week in Tech.* [Online] http://twit.tv/sn287.

19. FAQ. *Bitcoin Wiki.* [Online] https://en.bitcoin.it/wiki/FAQ.