



Exchange Digital Money using Bitcoin and Python

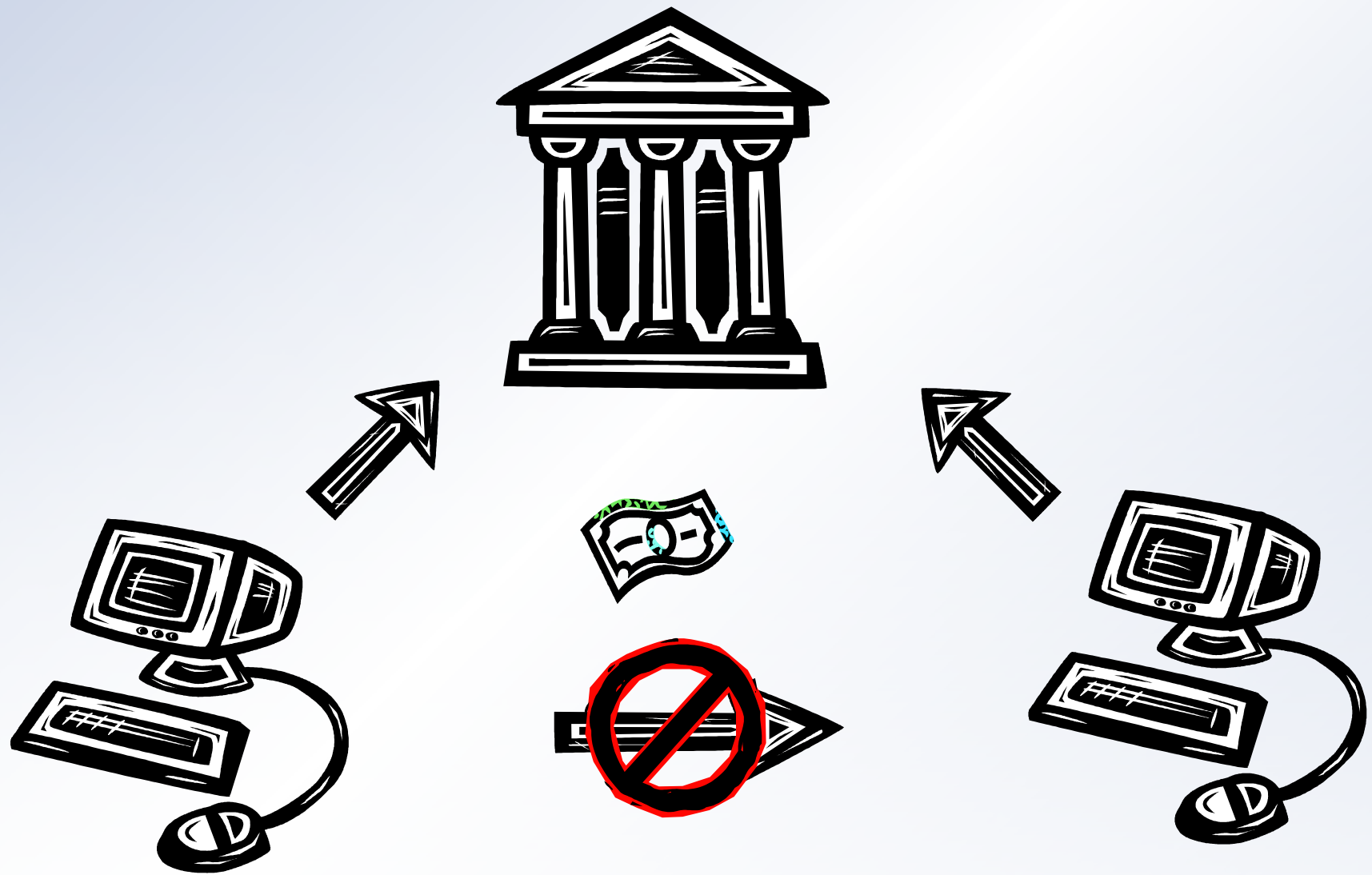
31 July 2011

David Steele
@dsteele
(+)dsteele@gmail.com

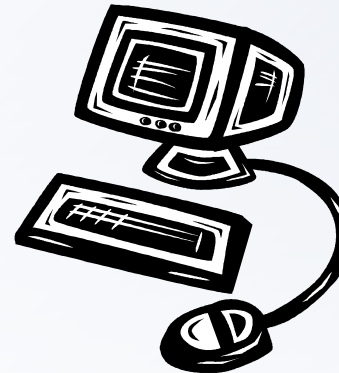
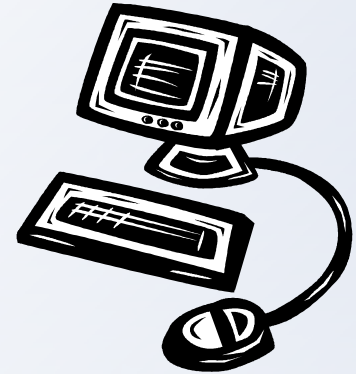
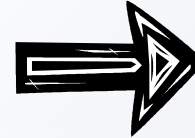
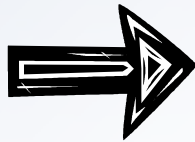
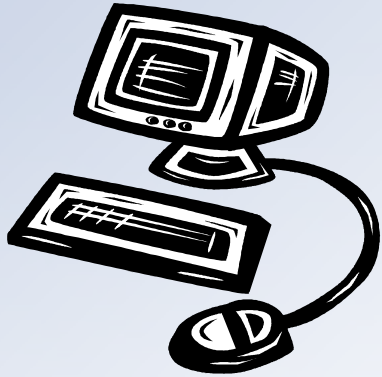
Resources

- Presentation
 - <http://davesteele.github.com/python-bitcoin-talk/>
- Software
 - {apt-get|yum} install git python-simplejson python-irc-lib pygame Django
 - git clone git://github.com/davesteele/python-bitcoin-talk.git











Block

1/3/09

t - 10 min

t



Transaction

Debit		Credit	
Address	AMT	Address	AMT
1LqRY...	17.21	avHR7...	5.00
		1LqRY...	12.21

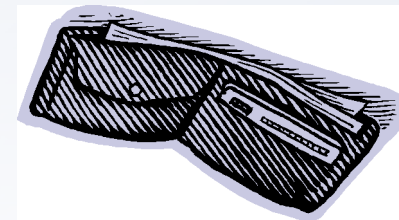
Address

1LqRY...

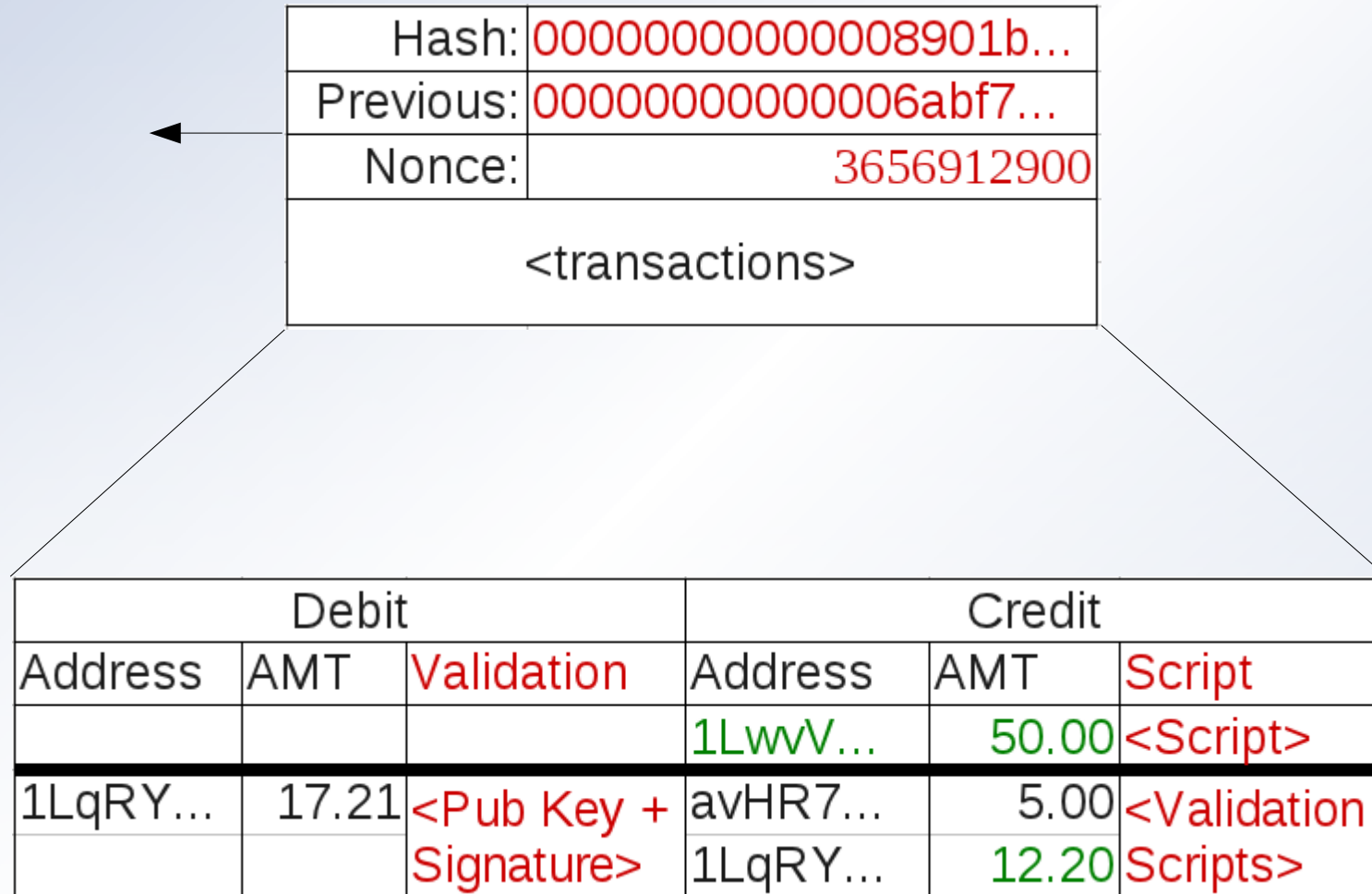
Public Key

Hash

Private Key

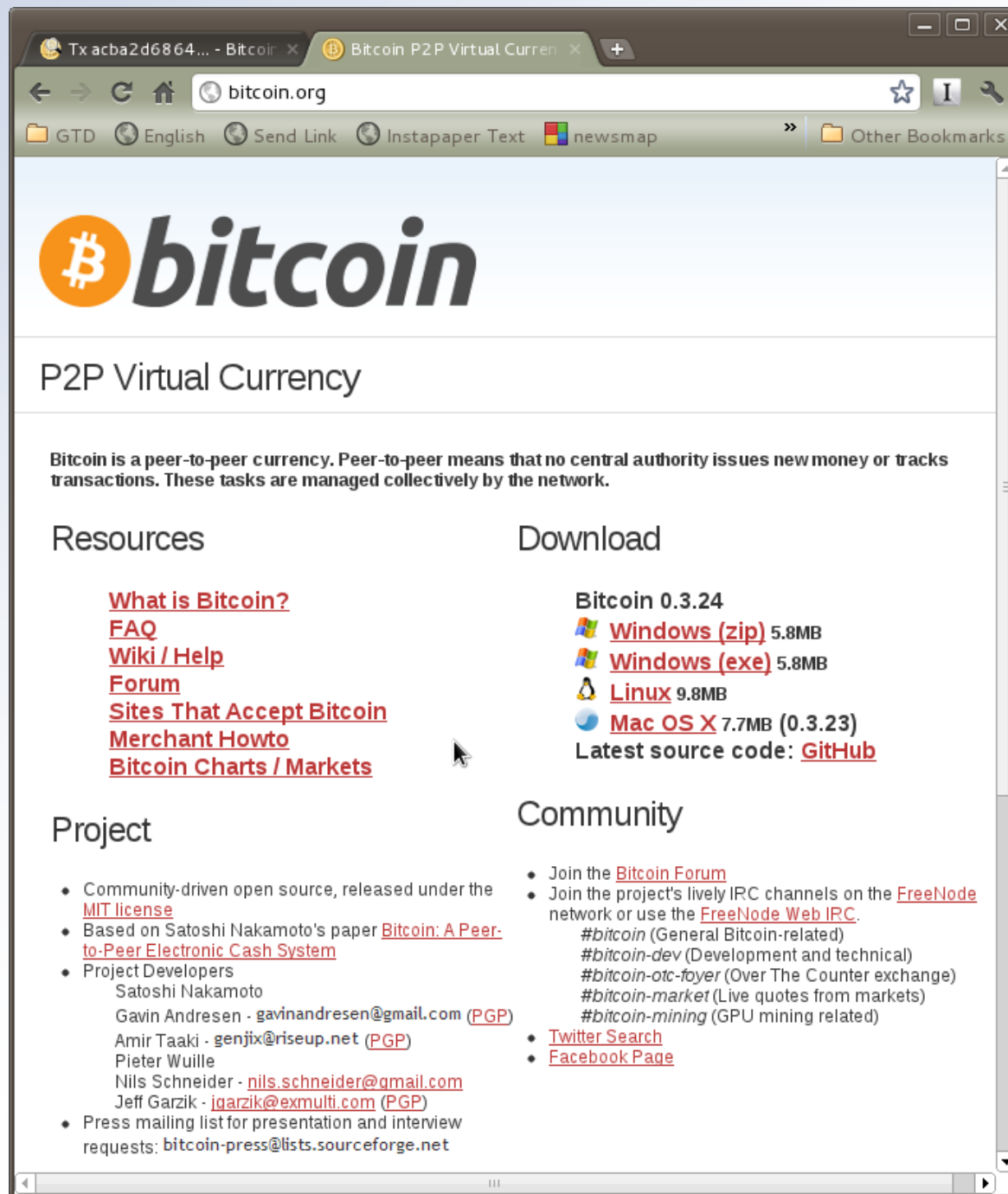


The Secret Sauce...

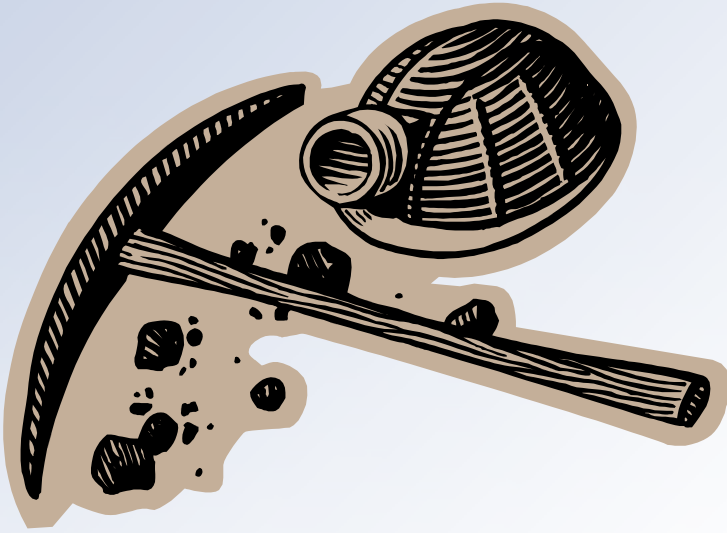


Terminology

- The Bitcoin ledger consists of a communally maintained chain of Blocks containing Transactions which record balance transfers between Addresses
- Addresses are derived from a hash of the public key for a public/private key pair stored in the private Wallet
- Transactions are Confirmed as they become embedded in the Longest Block Chain
- Addresses can be aggregated locally in the Wallet using text Labels, also called Accounts

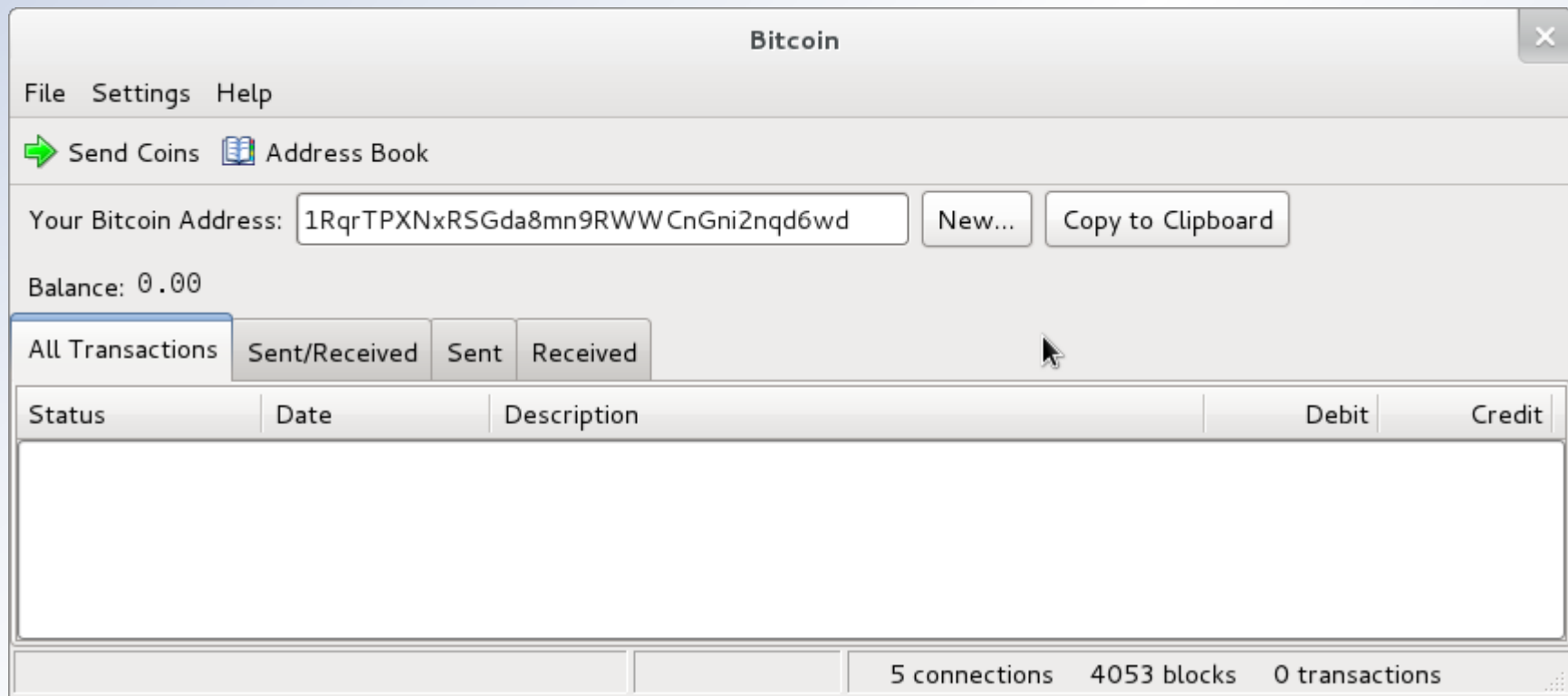


Getting Bitcoins



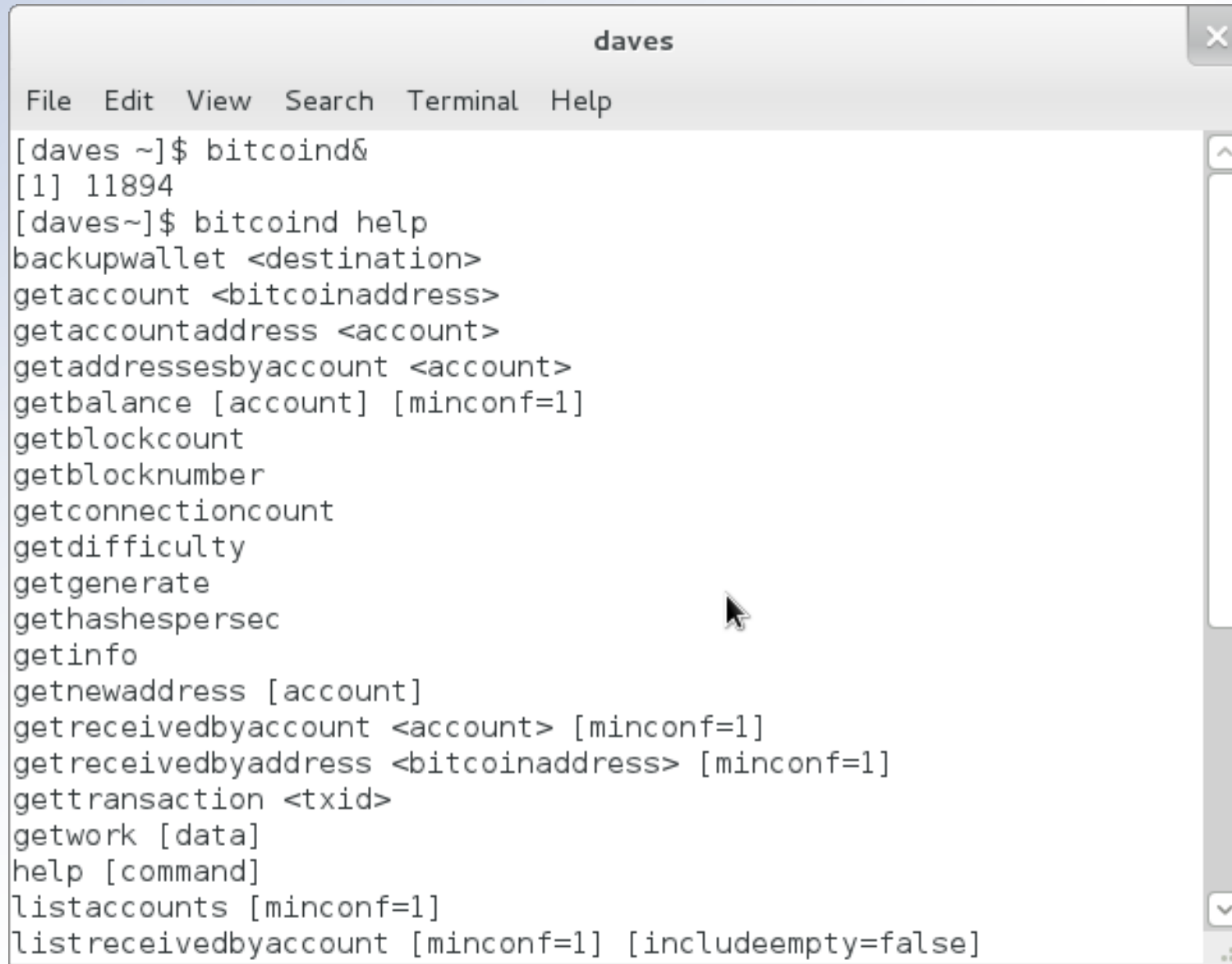
OR





Launch

bitcoind



```
daves
File Edit View Search Terminal Help
[daves ~]$ bitcoind&
[1] 11894
[daves~]$ bitcoind help
backupwallet <destination>
getaccount <bitcoinaddress>
getaccountaddress <account>
getaddressesbyaccount <account>
getbalance [account] [minconf=1]
getblockcount
getblocknumber
getconnectioncount
getdifficulty
getgenerate
gethashespersec
getinfo
getnewaddress [account]
getreceivedbyaccount <account> [minconf=1]
getreceivedbyaddress <bitcoinaddress> [minconf=1]
gettransaction <txid>
getwork [data]
help [command]
listaccounts [minconf=1]
listreceivedbyaccount [minconf=1] [includeempty=false]
```

bitcoind JSON-RPC using Python

- rpcuser and rpcpassword defined in
~/.bitcoin/bitcoin.conf
- Install python-bitcoinrpc
<https://github.com/jgarzik/python-bitcoinrpc>
- bitcoind running
- Ports 8332 and 8333 accessible
- API Calls defined
https://en.bitcoin.it/wiki/Original_Bitcoin_client/API_Calls_list

```
bitcoinclishort.py x
#!/usr/bin/python

from jsonrpc import ServiceProxy

PROXY = ServiceProxy( "http://me:mypassword@127.0.0.1:8332" )

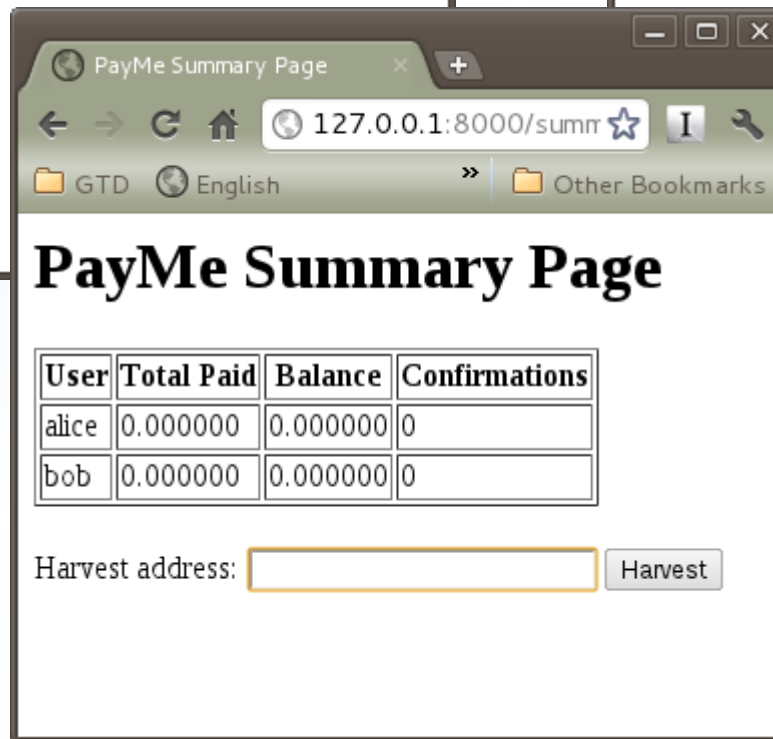
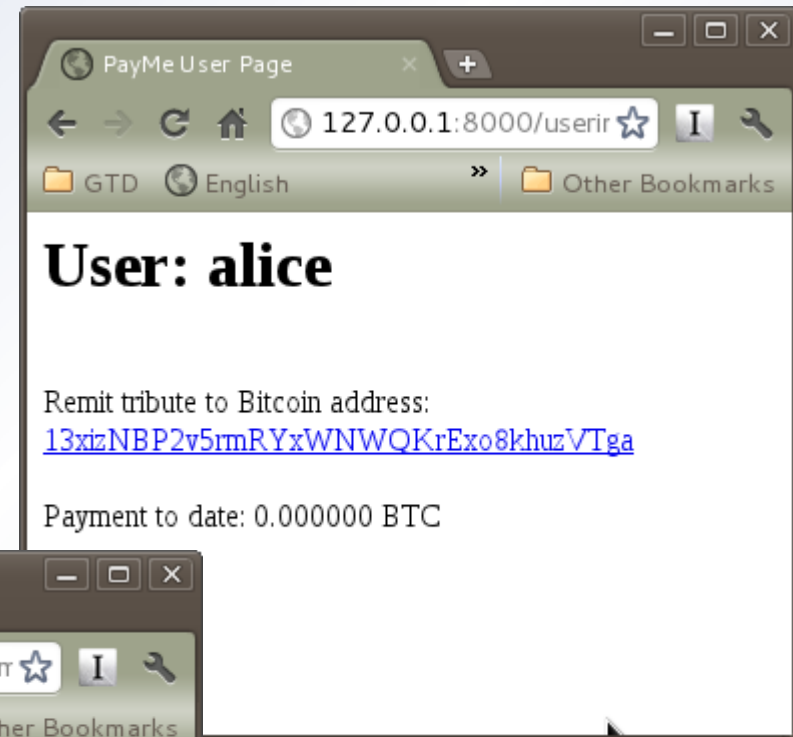
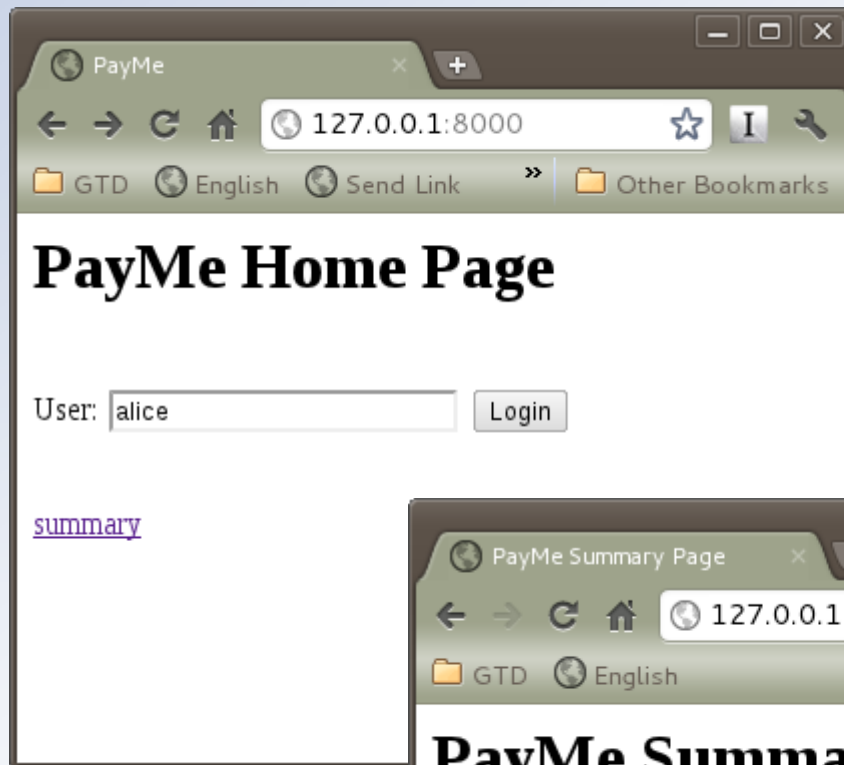
BTINFO = PROXY.getinfo()

print "BitCoin stats"

for key in BTINFO.keys():
    print "    %s; %s" % ( key, BTINFO[key] )

print "See https://en.bitcoin.it/wiki/Original_Bitcoin_client/API_Calls_list"
print "for the full list of functions available"
```

Example Django Website



"Bitcoin Watch" Monitor

The screenshot displays a Freenode IRC window for the channel #bitcoin-watch. The channel header shows the name and a list of users. The main area contains a stream of Bitcoin transaction logs, each starting with a timestamp [22:08] and a transaction ID (Txn). The logs show various transactions with their respective amounts in BTC and associated addresses. A terminal window titled 'daves:~/bitcointalk' is overlaid on the bottom right, showing the execution of a Python script. The script is a Bitcoin monitoring bot that connects to the Freenode IRC channel and monitors transactions. The terminal output shows the bot's initialization and the start of its monitoring routine.

```
FreeNode: #bitcoin-watch

IRC Edit Network Discussion View Help

freemode
#bitcoin-watch

[22:08] Txn e40a403f4da071ad32dc4102f6ae6dc9f94cb223c4c3917b074dc1abe1bbf04: 1Ab730xqvE7vtj4DPzdZqmuJswAv7kuaEK 0.16 BTC, 1HmKQ59haTq8raawLxRHKiKyzqAMeEr5FN 0.0395 BTC
[22:08] Txn 9cac293f057332fc67a844ae5abfc3837e7cb22d6a18df850269859ec19c8032: 15K629Uiy1fw2bDbHN2r7aTsdRuRYSDpVv 0.16 BTC, 1AVhSDaauZnZqpNgSFuDe2n7DcF2YDH0eq 0.0395 BTC
[22:08] Txn 2e9f214f065fa6237fc0cea175d4986658cdc79e4505bb67fed3c53c415ff071: 1GunEjMq1CXNjR1EnMDM29k8k6wxvVKuzx 40.19648305 BTC, 1B335NoZLZaStfwefJBCE17RqSfDsTN8j 0.17 BTC
[22:08] Txn 87a07b18df98568467d9bdb449169db5bf9e478d9248921b8176840b67e9c26d: 1Ab730xqvE7vtj4DPzdZqmuJswAv7kuaEK 0.16 BTC, 1GrPTHvTaiFRBQ4V7Y5GmUS3yG6HPkqQo 0.0395 BTC
[22:08] Txn bff43551f8215d051147f07623fce46d49a1e87333d44b34cb08ec723c960b00: 13RfoQwvEqZuR8yBwYhrEzNmM9ofNXXNRCT 1.00 BTC, 1En4vU4EX3FwC7p12U1tdztWkoMqzG2Fvt 0.32 BTC
[22:08] Txn f605b20be38997e9f7f04fd57730387a67b0086b081ec31e93a141eb79d62446: 16o11Cqs59HpoHABVe2S6W817QBTYxWkt 0.20 BTC, 12rZL2QqiB81RLf88TJknWXZiVAXaCbQ2Q 0.029 BTC
[22:08] Txn 1cc7c122505dff00f62b7c11db30bb321b7877cb9e379ac01ffa69c0cf2450fa: 1Ab730xqvE7vtj4DPzdZqmuJswAv7kuaEK 0.20 BTC, 1CSrTvr4u4yzrdNDzTGwUDqCBvAHjT6V4 0.0995 BTC
[22:08] Txn 39d2a312ebe5fb67eaa02abb4552b1f7f6bda1b902f19c8d49916a22e0996f69: 1GQ3EznNznhxqHATFr3ngeFKVtY5At0ZCU 40.071 BTC, 12xDHk1cbFmyFhd0qHdUhmKYHA9Z6r34cy 0.51 BTC
[22:08] Txn de51ec8ae43b47fb2fac07bf8d658c4a810f87ab7155cbf7807d8e01982aad3c: 1Ab730xqvE7vtj4DPzdZqmuJswAv7kuaEK 0.16 BTC, 1BmeFmqbGPPCNBL8DX96ud81VhtCdSchES 0.0395 BTC
[22:08] Txn 18b201b121b98e8d990b5f91bbfab361e14de6237b8ac9acebb1420fba17f6b5: 13LLyZwtFDDRPPhRXGAYffMYGTqYfa1Diq 0.10 BTC, 1BghuN3Tjew6PqPd69xDUChckrySxybX 0.039 BTC
[22:08] Txn 803428b206d584fd97f1be0eacbb2dcce417d87e3de8057f72c3b064473986c: 1GEZX1rEqWiVr7qbj3cZhoC9L1cZ9gBzvf 39.75041754 BTC, 18oipuHx35jpkqYw2QJH7npjUnR23YDzcf 1.01 BTC
[22:08] Txn f63c188cb88...
[22:08] Txn d7ab6026070...
[22:08] trade ExchngBCs:
[22:09] trade ExchngBCs:
[22:09] trade ExchngBCs:
[22:09] trade ExchngBCs:
[22:09] Txn 32e7708cf05...
[22:09] Txn 920c3cec950...
[22:09] Txn a72d33edcf8...
[22:09] BTC

92 Users daves
0.1s lag
```

```
daves:~/bitcointalk
File Edit View Search Terminal Help

#!/usr/bin/env python
""" module to monitor realtime Bitcoin transactions via the freenode
#Bitcoin-Watch channel - https://en.bitcoin.it/wiki/Bitcoin-Watch

requires irclib (repo)
and ircbot
(http://code.google.com/p/ircbot-collection/source/browse/trunk/ircbot.py?r=66)

pygame is used by the demo main routine"""

from ircbot import SingleServerIRCBot
from irclib import nm_to_n
import re
import random

class BitcoinWatchBot(SingleServerIRCBot):
    """ Bitcoin transaction watching bot. See main() for usage example """

    def __init__(self, channel="#Bitcoin-Watch",
                 nickname="bcbot",
                 server="chat.freenode.net",
                 port=6667,
                 realname="Bitcoin monitoring bot <webpage>"):
        16,7
        Top
```

Opportunity – URI Scheme Handler

bitcoin:1Lg7peCQCBRBRBsmZJ5MoXikuQ25oZ4voBit?amount=5X8&
label=Bitcoin%20Watch&
message=Donation%20for%20watch%20service



A screenshot of a Bitcoin payment dialog box. The dialog has a blue border and a white background. It contains the following text: "Bitcoin Payment" in bold, "To: Bitcoin Watch", "Amount: 5.0", "Memo: Donation for watch service", and "From Account: Alice" with a dropdown arrow. At the bottom, there are two buttons: "Cancel" (blue) and "Pay" (grey).

Bitcoin Payment
To: Bitcoin Watch
Amount: 5.0
Memo: Donation for watch service
From Account: Alice ▾

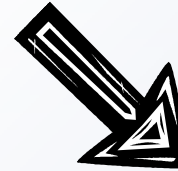
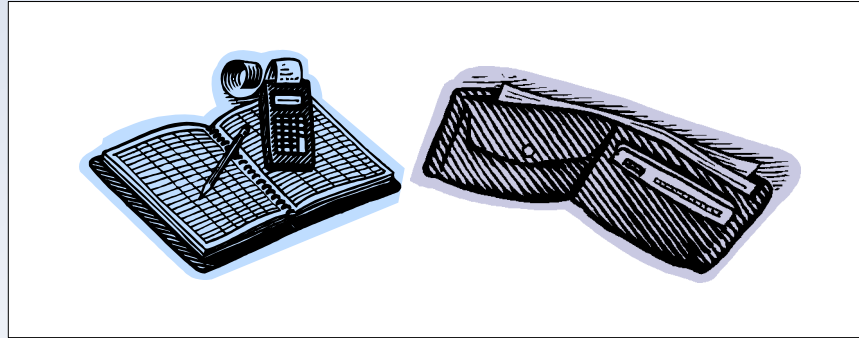
Cancel Pay

https://en.bitcoin.it/wiki/URI_Scheme

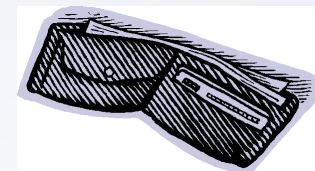
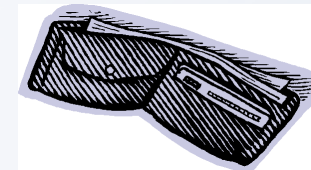
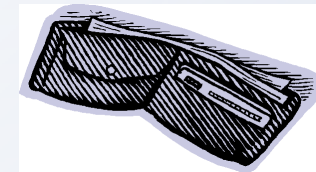
Vending



Opportunity – Wallet Apps



Different:
- Security Levels
- Users
- Use Cases



<http://gitorious.org/pycoin> - Bitcoin P2P Implementation

Python Resources

- dbdump.py – parse wallet file
 - <https://github.com/gavinandresen/bitcointools>
- pycoin – Bitcoin peer-to-peer implementation in Python 3
 - <http://gitorious.org/pycoin>

Aristotle's Qualities of a Good Money



Durable	✓	✓
Portable	✓	✓
Divisible	✓	✓
Intrinsic Value	✗	?

Bitcoin Links

- Market – Bitcoin Charts
 - <http://bitcoincharts.com>
- Plumbing – Block Explorer
 - <http://blockexplorer.com/>
 - Stats - <http://blockexplorer.com/q>
- Technical – Bitcoin Wiki
 - https://en.bitcoin.it/wiki/Main_Page
 - Addresses, Transactions, Signing, Blocks, Peer Protocol
- Community - <http://forum.bitcoin.org/>
- News
 - <http://www.bitcoinnews.com/>
 - http://www.****coins.com/

