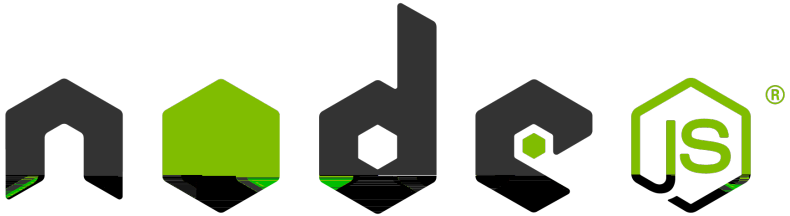
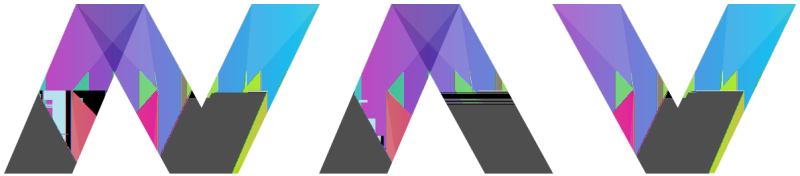




NAVAJO CC

WRITTEN BY CRAIG MACGREGOR

Introduction

About me

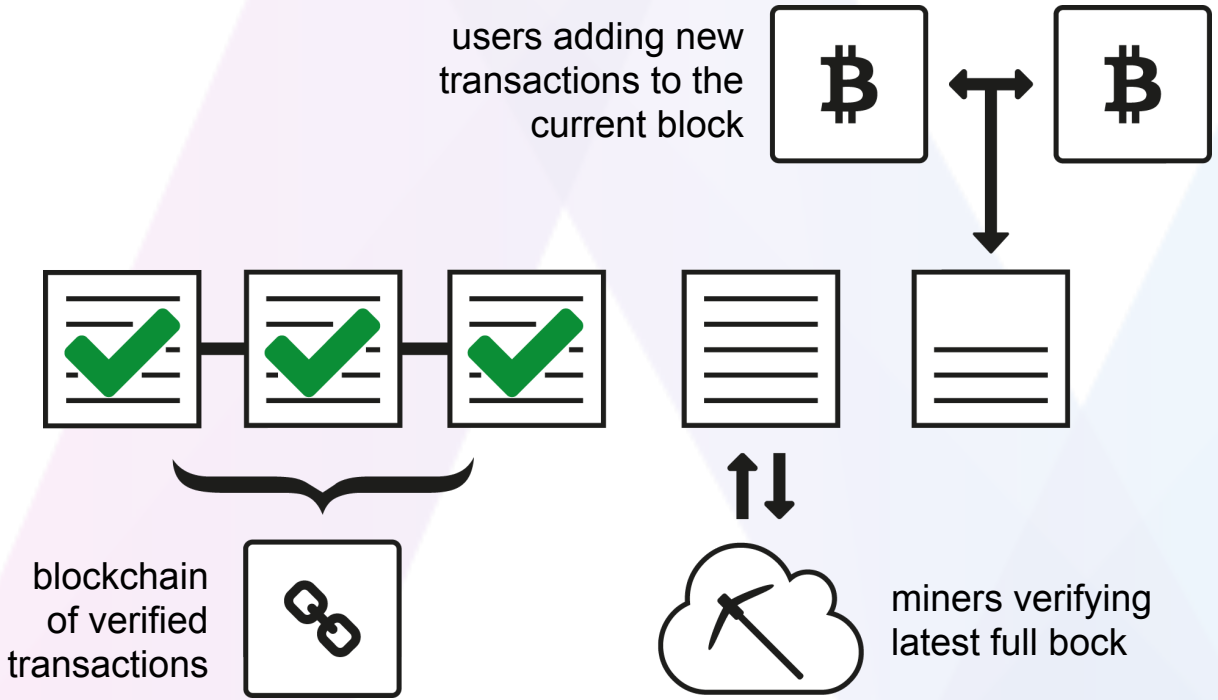
- ➔ Core developer of Navajocoin and its Anonymous System
- ➔ Developing cryptocurrencies since mid 2014
- ➔ Contract Developer and Cryptocurrency Consultant
- ➔ Web Developer for previous 10 years
- ➔ Blockchain technology enthusiast

Introduction

Contents

1. Blockchain basics
2. Navajocoin
3. NodeJS Demo
4. Navajo Anonymous Transactions
5. Blockchain technology and its applications

Blockchain Basics



- 10 minute block time
- Difficulty adjusted every 2016 blocks
- Network weight calculated by hashing power percentage
- Democratic agreement of completed block
- Autonomous and self sufficient network

Navajocoin

Comparing Navajocoin to Bitcoin

- ➔ Navajocoin uses Proof of Stake to calculate network weight
- ➔ 30 second block time
- ➔ Approximately 58 Million Navajocoin in circulation
- ➔ Diminishing mining rewards which flatten at 5% annual return

Navajocoin

Additional wallet features

- ➔ Chat system with encrypted private messages
- ➔ Market ticker to show current prices
- ➔ Staking calculator to show earnings
- ➔ Optional anonymous sending feature

NodeJS Demo

- ➔ Github of the Demo https://github.com/craigmacgregor/bitcoin-server
- ➔ Download Bitcoin Core - https://bitcoin.org/en/download
- ➔ Sync the blockchain
- ➔ Set the rpcusername and rpcpassword
- ➔ Test some rpc commands from terminal
- ➔ https://en.bitcoin.it/wiki/Original_Bitcoin_client/API_calls_list

NodeJS Demo

➔ Install angular-fullstack with yeoman http://yeoman.io

➔ Cool features of yeoman angular-fullstack
- REST API
- MongoDB Seeds
- Angular App scaffolding
- Bootstrap

➔ Branch: craigmacgregor/bitcoin-server/angular-fullstack

NodeJS Demo

- ➔ Install bitcoin RPC control - npm install bitcoin --save
- ➔ Check the RPC commands work from node
- ➔ console.log from /server/index.js
- ➔ Branch: [craigmacgregor/bitcoin-server/bitcoin-rpc]

NodeJS Demo

- ➔ Copied /server/api/thing to /server/api/wallet
- ➔ GET/api/wallets - listreceivedbyaddress(minconf=0, includeempty=true)
- ➔ POST/api/wallets - getnewaddress()
- ➔ Branch: [craigmacgregor/bitcoin-server/bootstrap-interface]

NodeJS Demo

- ➔ bower install angular-qrcode --save
- ➔ bower install angular-bootstrap --save
- ➔ Implemented QR Code of address into a modal
- ➔ GET/api/wallets/balance - getbalance(account=‘*’, minconf=0)
- ➔ Branch: [craigmacgregor/bitcoin-server/qr-codes]

NodeJS Demo

- ➔ Added bootstrap form to accept address and amount
- ➔ POST/api/wallets/send {address,amount}
  - validateaddress(address)
  - sendtoaddress(address,amount<float>)
- ➔ Update the balance
- ➔ Branch: craigmacgregor/bitcoin-server/send-coins

Navajo Anonymous Transactions

4.1

How it works

➔ Server randomly selected

➔ Destination encrypted and transaction sent to incoming server

➔ Server runs two coin daemons; Navajocoin and the Subchain

➔ Incoming Navajocoin transactions trigger Subchain transactions

➔ Navajocoin transaction randomised and sent to outgoing server

Navajo Anonymous Transactions

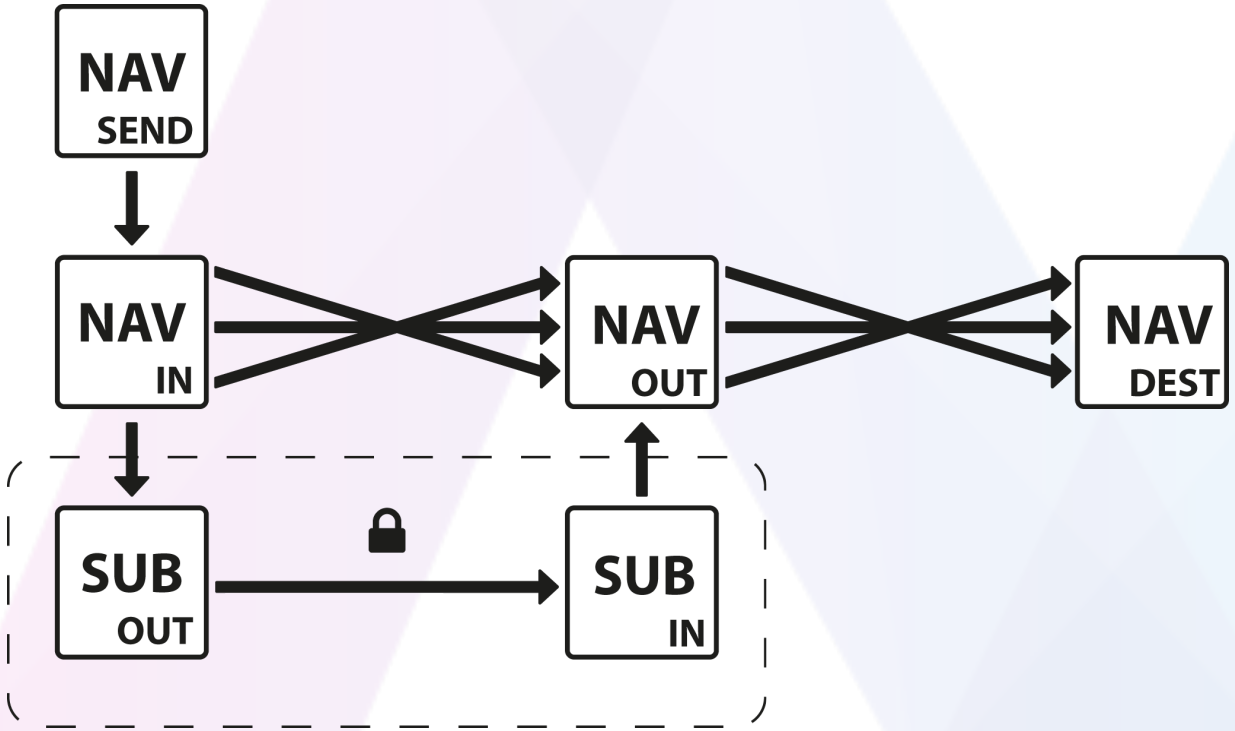
How it works continued...

➔ Incoming Subchain transaction triggers Navajocoin transaction

➔ Navajocoin mixed, randomised and sent to destination

➔ Subchain coins returned to origin server

Navajo Anonymous Transactions



- Encryption keys replaced every 24 hours
- No database
- Outgoing servers preloaded with NAV
- Outgoing transaction created before NAV is sent between servers
- NAV reaches destination within 5 minutes

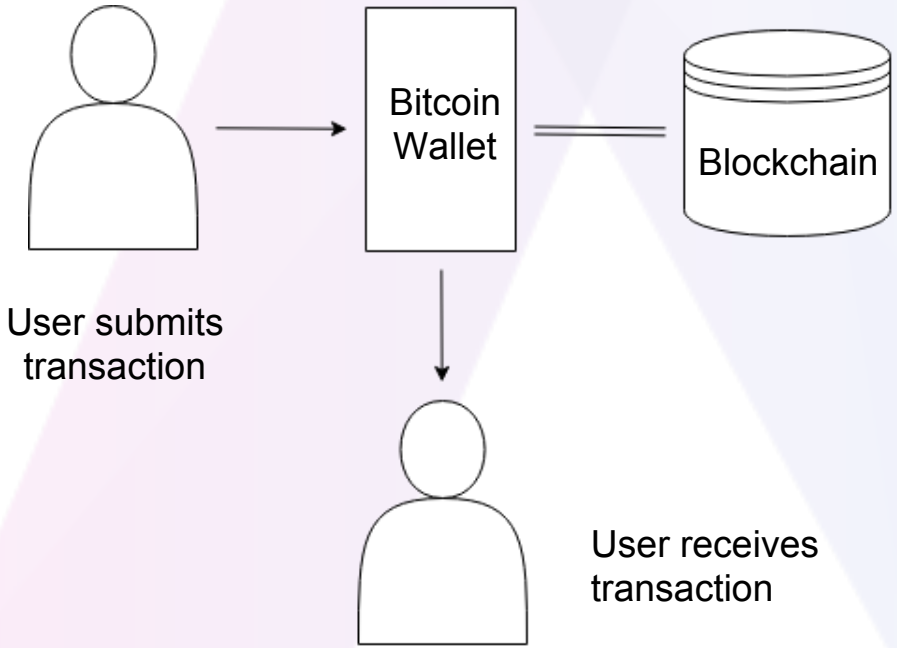
Navajo Anonymous Transactions

Technologies

- ➔ Linux, PHP and the Zend Framework (Rewriting in NodeJS)
- ➔ RPC interface between PHP and C++
- ➔ RSA Encryption of destination addresses
- ➔ SSL used for communication between servers
- ➔ No database!

Blockchain Technology Applications

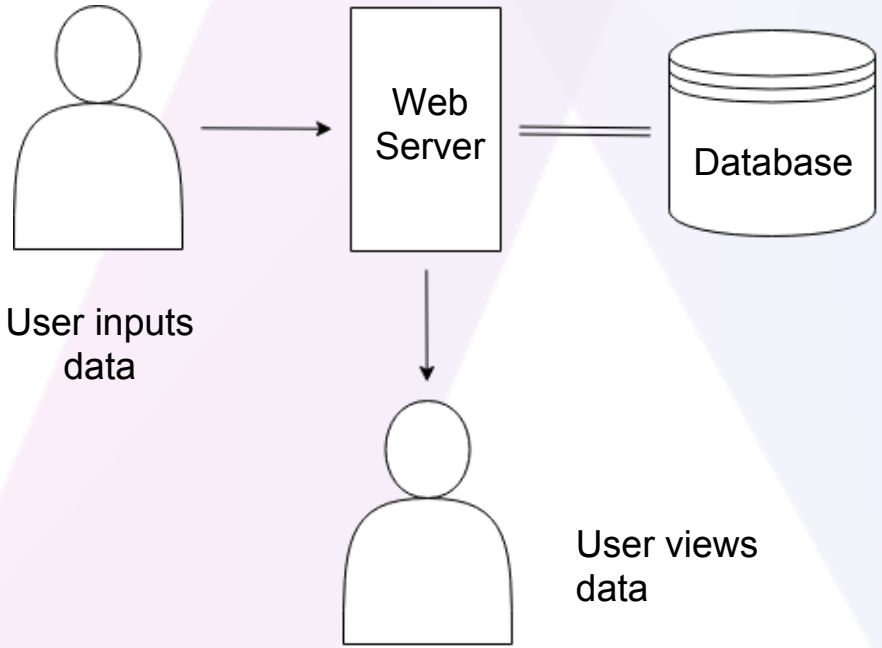
Basic cryptocurrency architecture



- User views their wallet
- Wallet balance loaded from blockchain
- User can submit a transaction
- Another user can receive transaction

Blockchain Technology Applications

Basic website architecture



- User views a website
- Website content loaded from database
- User can submit data to the database
- Another user can view the data

Blockchain Technology Applications

Blockchain application example

- ➔ Create a client which houses a bulletin board or forum
- ➔ Instead of submitting transaction data to the blockchain, users submit their messages
- ➔ Messages are validated by the client exactly like transactions
- ➔ Once verified the messages are distributed to the network

Blockchain Technology Applications

Advantages over traditional web architecture

- ➔ Decentralised
- ➔ Autonomous
- ➔ Democratic
- ➔ Virtually indestructible
- ➔ Almost impossible to manipulate

Blockchain Technology Applications

Technical Challenges

- ➔ No monetary value received from validating messages
- ➔ Network weight can’t be based on “work” or “stake”
- ➔ Storing large amounts of data like images or videos
- ➔ Legality of ownership of content and responsibility

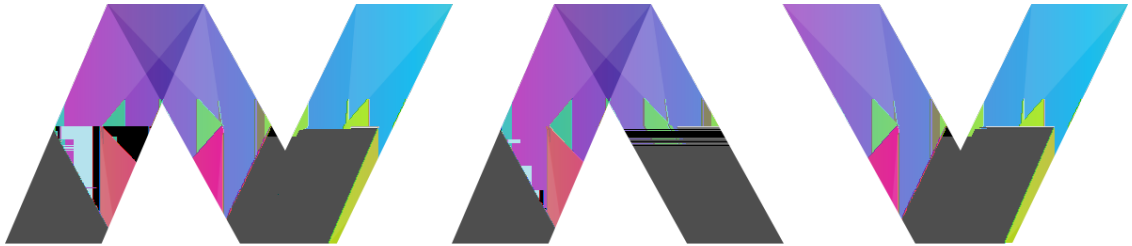
Blockchain Technology Applications

Navajo Anonymous Transactions

- ➔ Uses a blockchain to send data between servers
- ➔ One of the world’s first alternative usages of the blockchain
- ➔ Intermediate between financial and non financial blockchains
- ➔ Excited about the future and will continue to pioneer

Summary

- ➔ Blockchain architecture is versatile
- ➔ Uncensorable by governments or corporations
- ➔ Potential to restore freedom of speech to the digital age
- ➔ Probably won’t replace all website data storage
- ➔ Has an important role in the future of digital communication



NAVAJO CC

THE UNBREAKABLE CODE