NEO
smart economy

LEARN, THINK, CREATE

HAPPY NODES

NEO NGD Hackathon London 2018

Team F27

Shiran Chan, Wing Chan (~wy), Karlson Lee (~i25959341)

HAPPY NODES

21s
AVG BLOCK TIME

20s ago
LAST BLOCK SEEN

2,402,669
BEST BLOCK

Nodes

Online (36)

Asia (13)

Europe (9)

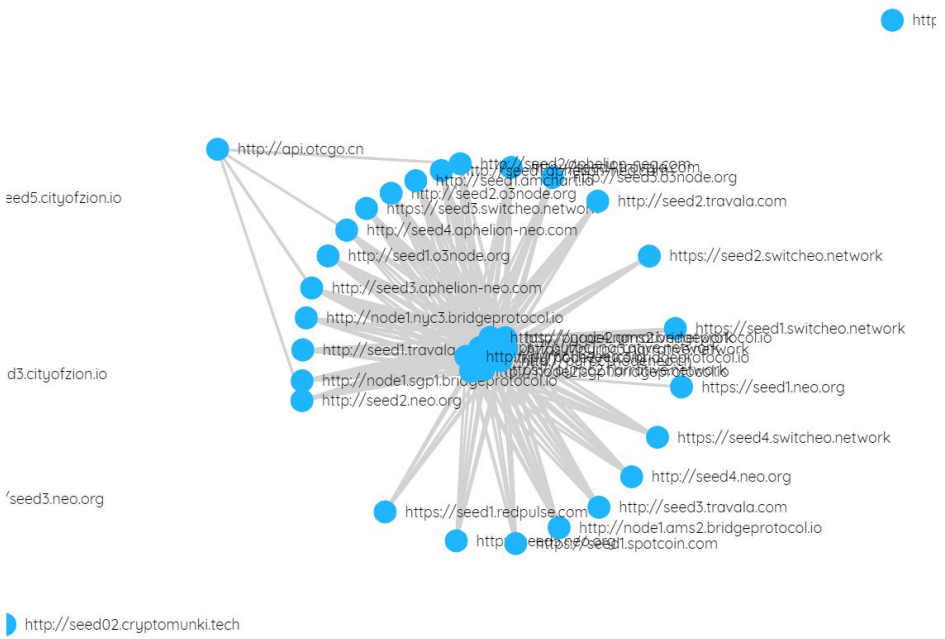
Americas (14)

Offline (9)

Asia (5)

Europe (3)

Direct Peers

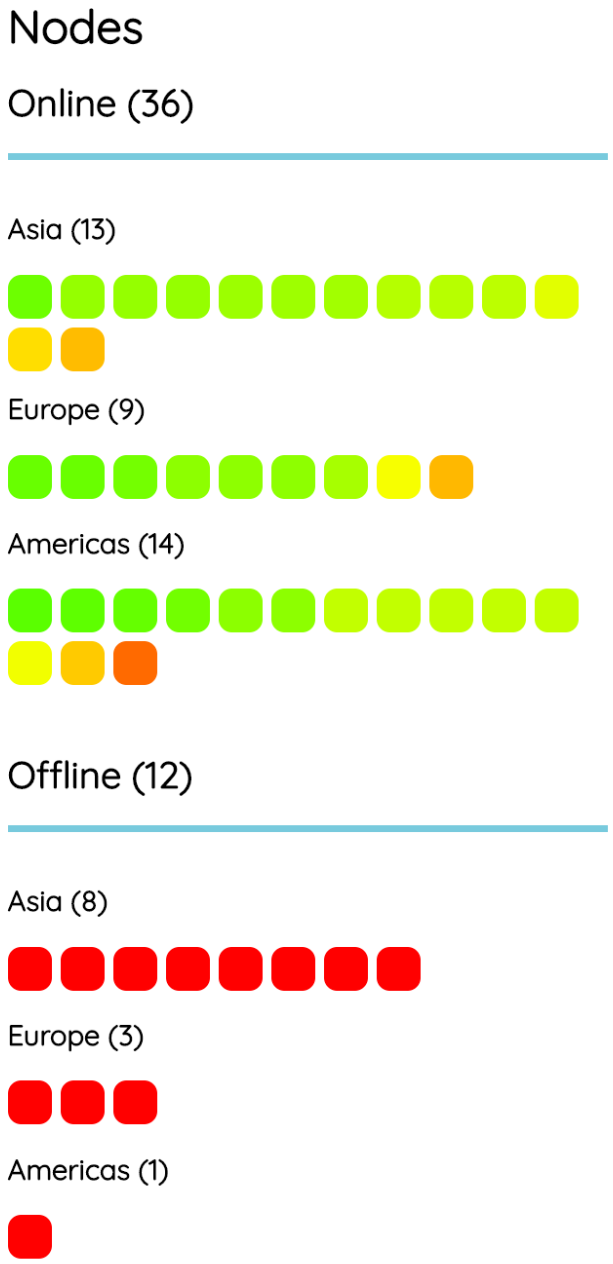


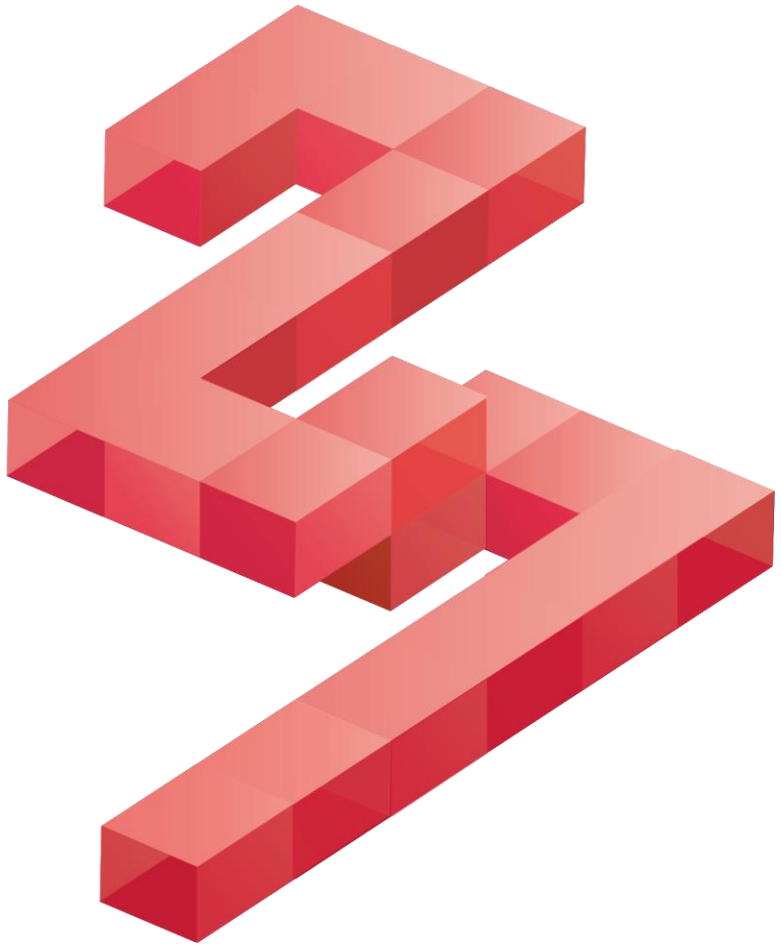
13 Unconfirmed Transactions for block 2402670

Showing top 10 (ranked by node agreement)

Transaction Reference	Node Count
fdd730218d96b7a920ffac61796dd9395d33203c2d1cb2a9ecb15bdb0b191c0b	31
0f2db24bf1fd93c4da75d390fc5733fdc5a8e9c04a966a30bc9195a4448519e0	31
57c25ce77664d4967ebc0dd0dc7857178ce75b8f961573eac2c40a22e42f8624	31
687a468d099d39eb64da4d3086402b2c36fcc8efe21ebd0ce28221e48c468795	31
a705e48ab47817409028d89bf5727399333cdefe60d879ac5b70df38f3096436	31
0c0b2e9c58af0a97418a6f4bce69351cc45c74ca0ea736ffa3ca72c2c3bd266b	31
319fb281bcd0e0880d6baa4bde656323c	31

Contents





About Us

We are Team F27

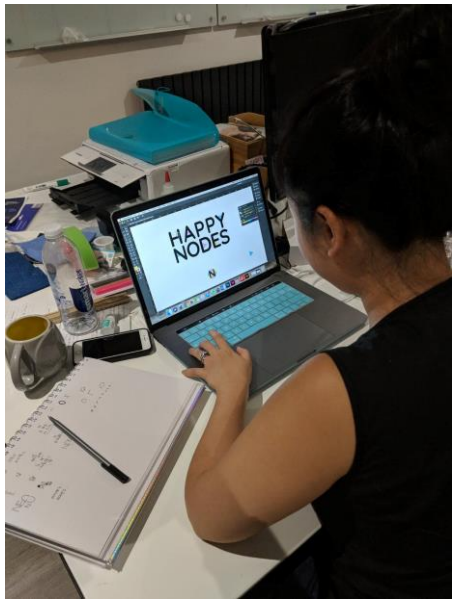
Shiran Chan – Graphic Design, Marketing
Wing Chan – Technology, Startup Investments
Karlson Lee – Technology, Cryptocurrency

We started F27 in 2018 to focus on three areas:

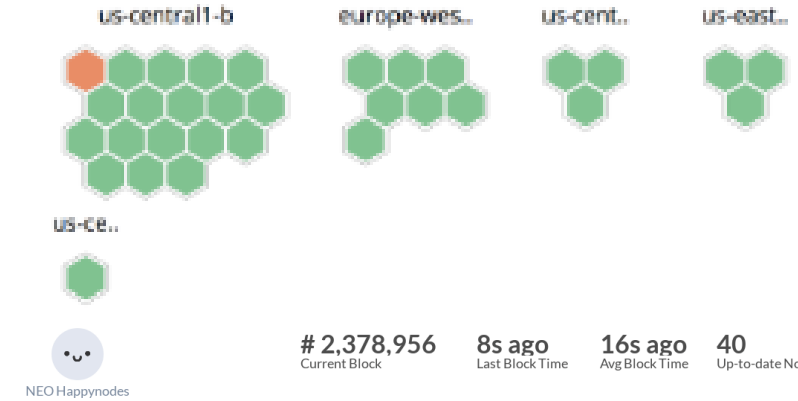
- Blockchain – technology & investments
- China – specifically population and cultural Diaspora
- Freelancers & On-demand economics

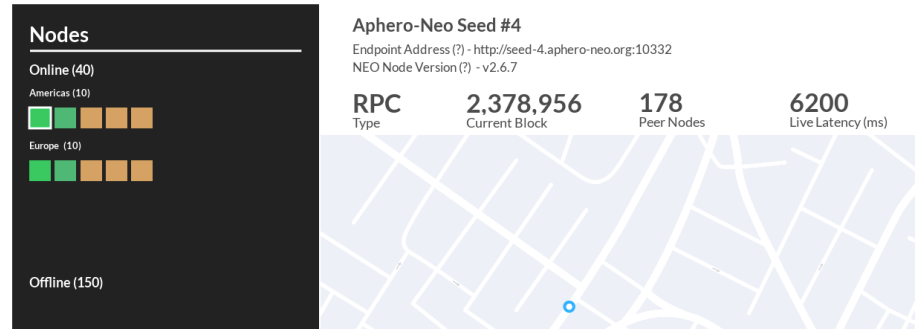
GitHub /F27Ventures

Pre-Competition Branding Session



HAPPY
NODES





HAPPY NODES

Our solution at a glance

We implemented a web interface and backend data repository for network state monitoring that focused on **innovation** and a **great experience for new users`**:

- Friendly, approachable interface that makes nodes in the network feel accessible (versus a data table)
- View the network state of unconfirmed transactions (not just how many, but for each transaction, how many nodes also share that pending transaction)
- Regional groupings of Nodes is a simpler way to show the geographic distribution of the network than a Map (generally perceived to be hard to navigate)
- A more nuanced health-score system than just Online (Yes/No)

110 Commits Later…

Architecture Overview

HAPPY NODES

Backend Processing

python™

PostgreSQL

Google Cloud Platform

API with Caching

node JS

PostgreSQL

Google Cloud Platform

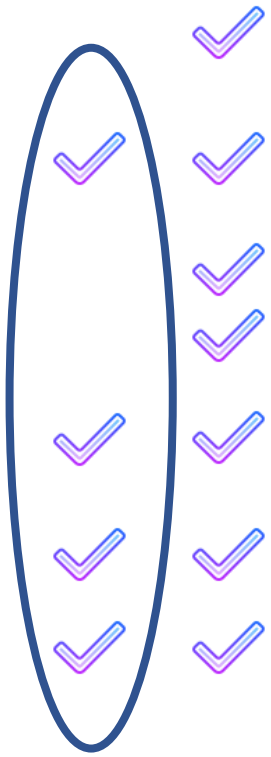
Web Front-End

node JS

React

kubernetes

Meeting the Functional Criteria



Information	Estimated Difficulty	Hint
P2P service status of NEO nodes	★★★★	Whether nodes provide P2P services via TCP/IP (10333), and P2P services via WebSocket (10334).
RPC service status of NEO nodes	★★★	Whether nodes provide JSON-RPC services on port 10331 and 10332
Stability of NEO nodes	★★★	If a node is offline for 10 minutes per day, the stability is `1 - 10/(60*24) = 99.3%`
Version information of NEO nodes	★★	http://docs.neo.org/en-us/node/cli/api/getversion.html
Current block height of NEO nodes	★★	http://docs.neo.org/en-us/node/cli/api/getblockcount.html
Current connection count of NEO nodes	★★	http://docs.neo.org/en-us/node/cli/api/getpeers.html
The number of transactions in the memory pool of NEO nodes	★★	http://docs.neo.org/en-us/node/cli/api/getrawmempool.html
Latency of NEO nodes	★★	Request delay for a user to a node.
Opening wallet on NEO nodes	★★	Opening a wallet on a node is extremely dangerous and should be monitored

Our innovations

Judging Criteria

Functionality
Implemented all expected features plus developed innovations on 4 of them

Easy to Use
Designed especially for people new to NEO and Blockchain Network monitoring – nodes are colour-coded based on their health status and grouped by territory. Clicking on them shows you more information

Stability
Only way to test this is to have run this for several days which we did! Our core solution has been running since Thursday, with incremental improvements to each layer since then.

Website Performance
More optimisation always possible but it can handle a large amount of traffic because most of the processing is done separately. Uses postgres for database backend with heightening API Caching to reduce server load

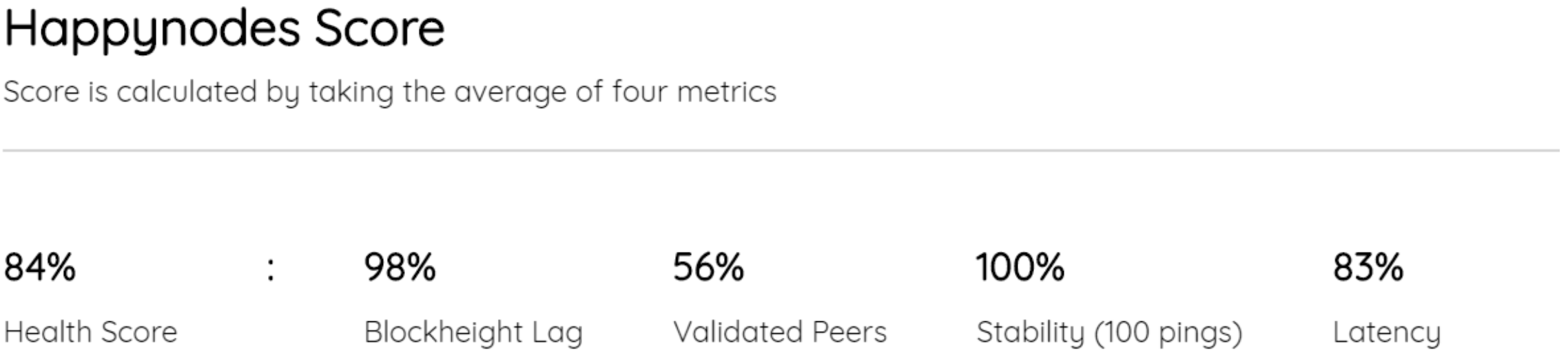
Interface Design
Design and branding of Happynodes was a key part of making the process fun and easy for users. NEO is a global and in the process of being decentralised, and showing that it is approachable is key.

Innovation #1 – Health Scoring

Current Approaches
City of Zion Monitor (neo-mon) uses a 4 category approach:

1. Up and Block lag < 3 – GREAT
2. Up and Block lag <= 1000 – WARNING
3. Up and Block lag > 1000 – NOT GOOD
4. Down – BAD / IGNORE

Their grouping uses just Up/Down and Block lag to determine the health of a node.



Innovation #2 – Publicly verifiable Connection Counts

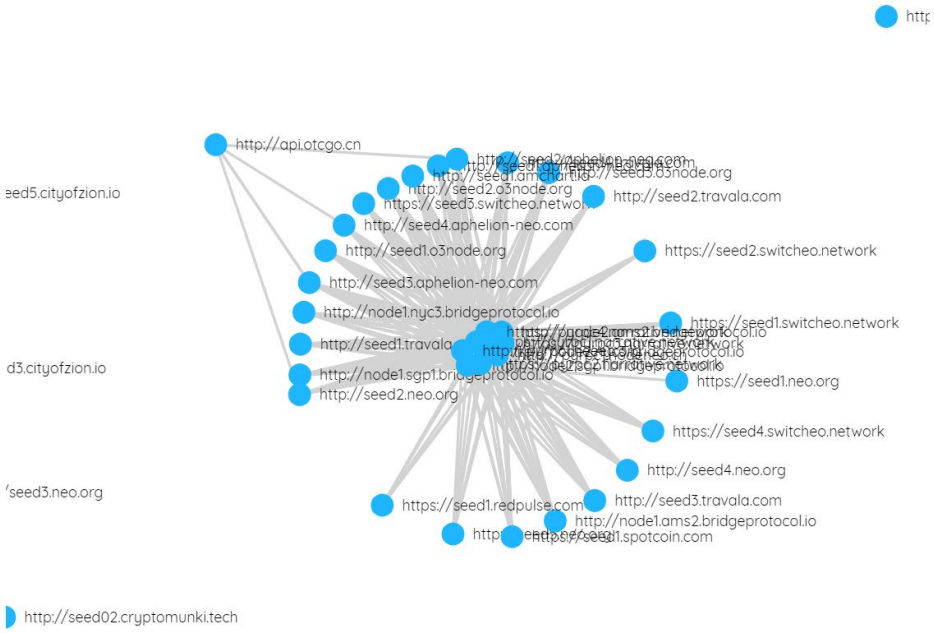
Current Approaches

City of Zion Monitor (neo-mon) simply uses the number of Connected nodes in the API response. However, many of those nodes don't correspond to known addresses or don't respond when pinged.

A more sensible approach is to only use the Connection Counts for connected nodes that can be queried as their own nodes.

We use this in our Health Scoring as well as exposing this to the users rather than just the raw connected count.

Direct Peers



Innovation #3 – Unconfirmed Transaction Explorer

Current Approaches

As far as we know, none of the trackers or monitors allow you to explore the network of unconfirmed transactions, opting instead to only show transactions that are confirmed.

This is normally fine because of NEO's superior block times (compared to Bitcoin and Ethereum) but over time, timing will matter.

Our Approach

We use the /getrawmempool API call on each node to get the list of transaction IDs and then merge the list together to figure out the unique set of transaction IDs as well as the transactions that are shared between them.

33 Unconfirmed Transactions for block 2402678

Showing top 10 (ranked by node agreement)

Transaction Reference	Node Count
fdd730218d96b7a920ffac61796dd9395d33203c2d1cb2a9ecb15bdb0b191c0b	42
0f2db24bf1fd93c4da75d390fc5733fdc5a8e9c04a966a30bc9195a4448519e0	42
17d6671f9511df644b6f81fa9d13a311a8e947276f2442bc35cb5fff991647b5	42
278b56b702cb782c53f50d9c5d655755edc2cc028f5ce4df79a11ae6186052ae	42
a705e48ab47817409028d89bf5727399333cdefe60d879ac5b70df38f3096436	42
ad4ecbb512dc05ebf68dafbbd4fba080d38a3e2ba0a3a18452ccf24685239b7a	42
b44471c6858b0d6340f2843f19d016ab3cfeb17298a159962b264ca102defdb1	42
b469a3523545ae72831c1e5cb904b75e0bdf1f1435b5e84dc76e4cfdd59f18c0	42

Our learnings (1)

This was a great opportunity to get deeper into the fundamentals of how the NEO network functions, here are some maybe non-trivial observations:

- The RPC call to getpeers returns back three lists:
    - Connected
    - Unconnected
    - Bad
  - We expected Connected to contain addresses that we can link back to other peers, however, we encountered some issues:
    - Many of the addresses were local ipv6 addresses so we couldn’t ping them
    - Many of the addresses didn’t return any response and don’t map to known host addresses
  - We wonder if whether they should be using the address (in ipv6 format) or if they should use the http/https address instead?

Our learnings (2)

- We wanted to investigate the Validator nodes to graph this or show it varied over time but it was always the same node. Hoping this changes over time as we decentralise

- Existing NEO investigative services allow you to search by transaction, block or address but they generally disregard pending transactions – it was interesting to look at these unconfirmed transactions and how they propagated.

Easter Egg

Running /getnewaddress on the aphelio node reveals some private information:

{
"jsonrpc": "2.0",
"id": 1,
"error": {
"code": -32601,
"message": "Method not found",
"data": " at Neo.Network.RPC.RpcServer.Process(String method, JArray _params) in **/home/solinsky/repos/aph-server**/neo/Network/RPC/RpcServer.cs:line 320\n at Neo.Network.RPC.RpcServer.ProcessRequest(HttpContext context, JObject request) in /home/solinsky/repos/aph-server/neo/Network/RPC/RpcServer.cs:line 405"
}}