# **Transaction Tracking**

# **1 Basics**

A table with an overview of the support status and applicability.

|  |  |
| --- | --- |
| Status: | **Supported** |
| Architecture(s): |  |
| Component(s): | Blockchain Core, NodeJS, additional web languages |
| Hardware: | *n/a* |

# **2 Overview**

Transaction tracking is an important aspect of any blockchain both for the user and the community as a whole. This open ledger should be accessible to anyone at any time, typically through the use of a ‘Blockchain Explorer.’ which utilizes APIs to access the blockchain and display information to the user in an easily digestible manner.

# **3 User details**

Transaction tracking will be accessible through the user’s wallet (in an abridged capacity), through the CLI, as well as through the Blockchain Explorer (diviscan).

**User wallet**

* See last 10 transactions made by wallet (default)
* See full list of transactions by request
* Search transactions by date, hash, address (username)

**Blockchain explorer**

* Block times / height
* Last block
* USD, BTC pricing pairs
* Search
  + Tx hash
  + Address (username)
  + Block

# **4 Technical details**

Once the blockchain is complete, APIs will need to be written to communicate with the blockchain programmatically. This allows for web applications like the blockchain explorer and user-facing transaction tracking to exist.

**NodeJS pros**

Utilizing node allows for the backend, cross-platform communication that can be easily built out to be a robust, programmatic method for accessing transactions on the blockchain.

**5 Limitations**

Blockchain and web developers will need to come together to develop this technology as it will ultimately be tied to the blockchain core. This causes limitations in that there is no separation of concerns and time requirements need to be aligned between parties (potential scheduling conflicts). Once the software is developed, however, the blockchain team can once again be separated from the web team as they will focus on testing.

# **6 Testing**

Testing on this feature should be done by the software development team to encourage a separation of concerns between blockchain and software development. It will also allow for the software team to begin development on a simple, yet robust blockchain explorer, which is necessary to access larger exchanges.

Tests will include API calls using various technologies to test for speed and accuracy and will require a testnet sidechain.

# **7 Areas for improvement**

Stretch goals for this feature include an open-source SDK for other developers to utilize and build useful tools using the base technology. Examples include WordPress plugins, more feature-rich blockchain explorers, and more.

# **8 Known issues**

None known at this time.

# **9 References**

<https://etherscan.io/> - Blockchain explorer for Ethereum

<https://blockchain.info/> - Blockchain explorer for Bitcoin