Towards a real-time asset-backed cryptocurrency

Michael Christopher Bell (BSc, Stir)

February 2017

Dissertation submitted in partial fulfilment for the degree of
Master of Science in Computing for Financial Markets

Computing Science and Mathematics
University of Stirling

Abstract

Problem: We tackled several important flaws of cryptocurrencies by demonstrating a potential solution as a 'proof of concept' (POC). This solution was needed as almost all crypto-currencies are based on proof-of-work blockchain technology, have highly volatile market valuations, with each financial transaction requiring between ten minutes to an hour to clear. Similar to fiat-currencies (since the 'gold-standard' was abandoned), almost all crypto-currencies are void of tangible financial value. 'Proof of work' blockchains have also been widely criticised for the wasteful energy costs involved in the 'mining' process. The above flaws are important barriers to popular adoption which we believe is a challenge worthy of novel innovation.

Objectives: Our most important objective was the design, development and implementation of a performant, cross-platform, real-time, web app for near instantaneous financial transactions in a democratically determined asset-backed cryptocurrency.

Methodology: We set out to solve the problem by selecting 'Meteor', a bleeding-edge open-source web application framework (built upon 'NodeJS') in order to rapidly create an extensible real-time web app prototype. We adopted 'React' as our 'View' engine, and selected Steller's 'proof of consensus' blockchain technology as the financial transport layer due to its excellent documentation, helpful team and ethical values. While many open source packages were considered and tested, we always favoured simplicity over complexity, and at all times avoided 're-invention of the wheel'.

Achievements: We managed to complete the majority of our use-cases initially set-out in our project proposal. We gained significant in-depth working knowledge and experience of project management, systems design, full-stack software development utilising state-of the art frameworks such as Meteor and React. This required significant coding in Java, JavaScript, HTML, CSS, and API integrations in a number of libraries. We solved unforeseen technical integration issues concerning Stellar.org's JavaScript API (on Windows 8.1) by communicating with Stellar.org's core development team on Stellar.org's slack channel and discussing solutions. This resulted a highly secure Java backend sever coupled with a fast client-facing Meteor/JS server in which we achieved a clean 'separation of concerns' that should aid the process of growing the POC into a marketable application of popular social interest.