Semantic Blockchain

Software Requirements Specification

Version 0.1

May 13, 2016

Team Members:
Firas Kassawat, Ernane Luis, Matthew English

Mentor:
Prof. Dr. Maria-Esther Vidal

---

1. Introduction

We live in an age where the vast majority of blockchain explorers [1,2] export the raw data of the transaction network to a relational database, we aim to export it to a graph database defined according to the RDF specification. Our proposed project is to create, to the best of our knowledge, the world's first *'Semantic Blockchain'*.

1.1 Purpose

The purpose of this document is to set forth a clear prospectus on the implementation details of our software framework.

1.2 Scope

This project will produce a graph database (using Semantic technologies) to represent the Bitcoin blockchain, which will facilitate exploratory and visual analytics in addition to subgraph pattern matching as well as user-defined reporting.

1.4 References

[1]https://blockchain.info/
[2]https://blockexplorer.com/
[3]http://static1.squarespace.com/static/55295fd8e4b0719b1e03c0f5/t/557dea05e4b00a74542c192e/1434315335866/
[4]http://n-o-d-e.net/post/115030545546/how-to-build-a-bitcoin-node-on-the-raspberry-pi-2

[5]https://github.com/EIS-Bonn/BLONDiE/

1.5 Overview

This document is organized as by the requirements, the requisite hardware, and the envisaged use cases.

2. General Description

The initial system requirements are herein laid out, for changes see Section 5.

2.1 Product Perspective

This product is a “Semantic Blockchain” and the associated exploration framework.

2.2 Product Functions

The software will perform the functions of…

2.3 User Characteristics

Users of the system will have a basic familiarity with blockchain technologies, visual analytics and common software paradigms.

2.4 General Constraints

The implementation of the system should be operational.

2.5 Assumptions & Dependencies

This project is dependent on the Bitcoin blockchain and the tools that are slated to serve as dependencies.

3. Specific Requirements

What follows is a general outline of the specific system requirements.

3.1 External Interface Requirements

As the Bitcoin blockchain constitutes approximately 40 GB of information, we will require server space of roughly 60 GB.
According to our research [4] it should be possible to build the necessary machine using a

Raspberry Pi at a cost of not more than €100.

3.1.1 User Interfaces

Users of the system will need only common personal computer equipment or smartphones to interact with the system.

3.1.2 Hardware Interfaces

Users can interoperate with the system by means of a standard monitor, mouse/trackpad, and keyboard, etc..

3.1.3 Software Interfaces

Browser based solution users can navigate to on the web.

3.1.4 Communication Interfaces

Standard input output mechanisms.

3.2 Functional Requirements

This section describes specific features of the software project.

3.2.1 Semantically Describe the Properties of a Transaction

- Using RDF
- Define a vocabulary (BLONDiE) [5]

3.2.2 Ability to Link to Other Data Resources

Since our aim is to make the blockchain information into 5-start open data, for fill out the 5-start requirements we need to have the capability to link our data do another resourcer to the linked data cloud. Thus, our data has to:

- 4-stars) Use URIs to denote things, so that people can point at your stuff
- 5-stars) Link your data to other data to provide context

For that requirements we are going to use tools like SILK,The Linked Data Integration Framework, which it will give us the ability for integrating heterogeneous data sources. The primary uses cases of Silk include:

- Generating links between related data items within different Linked Data sources.
- Linked Data publishers can use Silk to set RDF links from their data sources to other data sources on the Web.
- Applying data transformations to structured data sources.

3.2.3 Ability to Discover Patterns

One of the primary aims of this software is to map the community structure of the blockchain network, i.e. the graphs representing the connections among the counterparties tied to a transaction (node). Initially we plan to introduce this functionality on the Bitcoin blockchain but our approach will be sufficiently generalizable such that the possibility to extend the service to other “altcoins” will be feasible.

Toward this end, we seek to design a new technique to efficiently build and cluster all the actors within the blockchain ecosystem. Analogous experimental findings, viz. “Ego-nets”, have been quite compelling, at a microscopic level it is demonstrable that high quality “communities” can be detected. Leveraging this concept we will engineer novel features for proprietorship identification based on co-occurrences of identical node across various exchange communities. This feature will be computed across this large scale graph through an analysis of the counterparty profile associated with each node.

3.2.3 Provide Visualization

Visual analytics is an effective mechanism for utilizing the powerful human visual cortex in the process of deriving meaningful insights from complicated datasets.


```
{
    "last": { "@link": "QmZ..." },
    "transaction": { "@link": "QmA.." } // { "action:" ... }
    "tick": 587174
}
```


Figure 1. Raw blockchain data

The benefits of graphically representing the network are demonstrated in Figure 2. wherein we can see a subset of prototypical subgraph patterns rendered suitable for a quick visual comparison more efficiently than would be possible with a large cadre of files in the style of Figure 1.

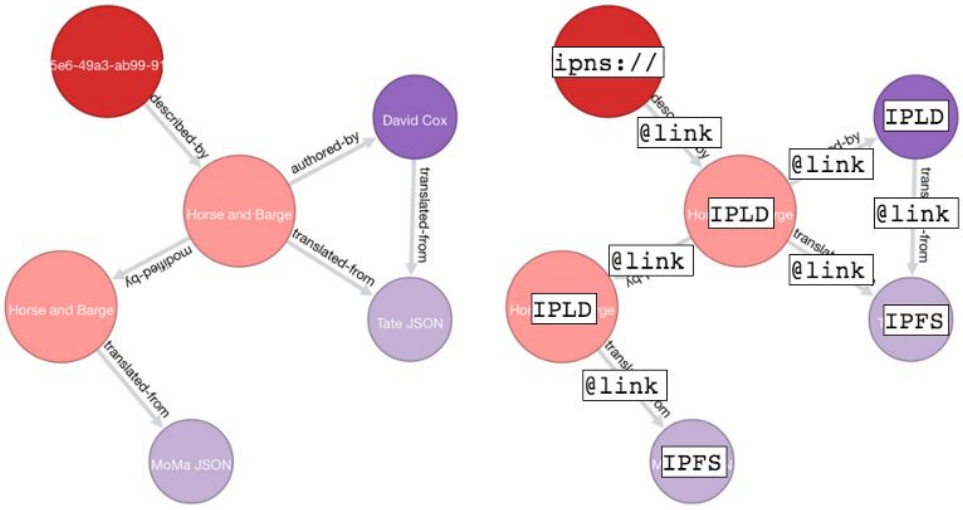


Figure 2. Visualization of a subgraph pattern comparison

3.3 Use Cases

3.3.1 User-generated reports

The system shall be capable of outputting reports that satisfy the demands of a user-defined query, as in Figure 4.

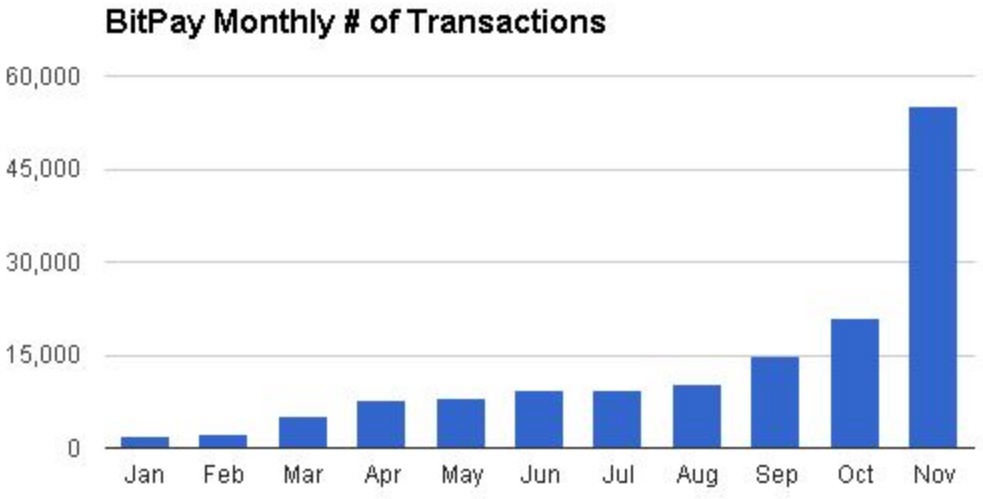


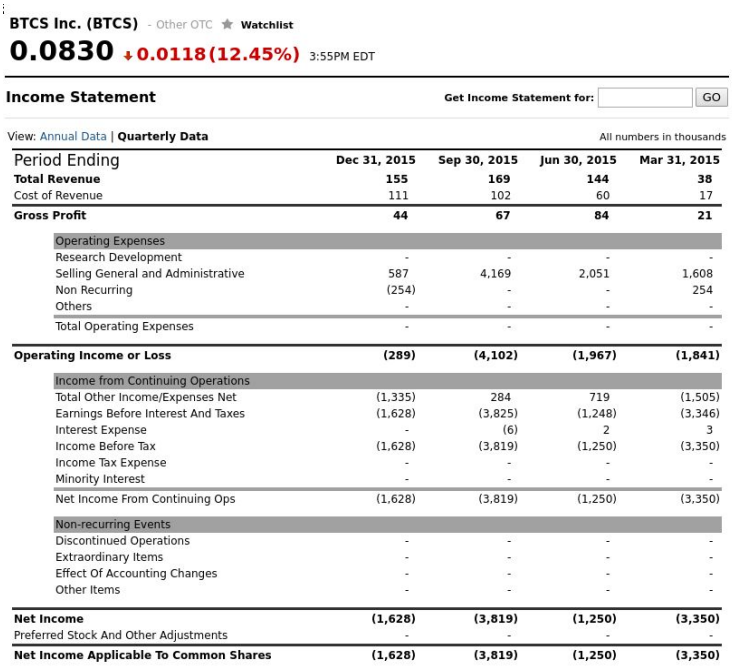
Figure 3. Example of report generation feature

3.3.2 Subgraph Pattern Matching

Bitcoin wallets can be associated based on shared characteristics and attributes, for example, given the pseudo-anonymity of the network- if it is noticed that a certain address sends and receives transaction of a certain amount at a certain time with some regularity this might indicate that this wallet is somehow related to wallets that behave similarly, for instance, the identification of wallets that belong to popular exchanges, wallets that belong to gambling websites, and so on.

3.3.3 Generating Financial Reports

Institutions and individuals with an interest in Bitcoin will employ our software to generate reports based on a series of pre-specified criteria.



BTCS Inc. (BTCS) - Other OTC ★ **Watchlist**

0.0830 ↓**0.0118 (12.45%)** 3:55PM EDT

Income Statement — **Get Income Statement for:** GO

View: Annual Data | **Quarterly Data** — All numbers in thousands

Period Ending	Dec 31, 2015	Sep 30, 2015	Jun 30, 2015	Mar 31, 2015
Total Revenue	**155**	**169**	**144**	**38**
Cost of Revenue	111	102	60	17
Gross Profit	**44**	**67**	**84**	**21**
Operating Expenses				
Research Development	-	-	-	-
Selling General and Administrative	587	4,169	2,051	1,608
Non Recurring	(254)	-	-	254
Others	-	-	-	-
Total Operating Expenses	-	-	-	-
Operating Income or Loss	**(289)**	**(4,102)**	**(1,967)**	**(1,841)**
Income from Continuing Operations				
Total Other Income/Expenses Net	(1,335)	284	719	(1,505)
Earnings Before Interest And Taxes	(1,628)	(3,825)	(1,248)	(3,346)
Interest Expense	-	(6)	2	3
Income Before Tax	(1,628)	(3,819)	(1,250)	(3,350)
Income Tax Expense	-	-	-	-
Minority Interest	-	-	-	-
Net Income From Continuing Ops	(1,628)	(3,819)	(1,250)	(3,350)
Non-recurring Events				
Discontinued Operations	-	-	-	-
Extraordinary Items	-	-	-	-
Effect Of Accounting Changes	-	-	-	-
Other Items	-	-	-	-
Net Income	**(1,628)**	**(3,819)**	**(1,250)**	**(3,350)**
Preferred Stock And Other Adjustments	-	-	-	-
Net Income Applicable To Common Shares	**(1,628)**	**(3,819)**	**(1,250)**	**(3,350)**

Figure 4. Raw blockchain data

3.5 Non-Functional Requirements

Non-functional requirements may exist for the following attributes. Often these requirements must be achieved at a system-wide level rather than at a unit level. State the requirements in the following sections in measurable terms (e.g., 95% of transaction shall be processed in less than a second, system downtime may not exceed 1 minute per day, > 30 day MTBF value, etc).

3.5.1 Performance, Portability & Maintainability

- Provide reports of transactions with a chosen criteria
- Provide search and track operations on transactions.

- Provide local on the fly processing of transactions

3.5.2 Access Control

As we know the blockchain for bitcoin is available to anyone, however this data just contains simple information about the transactions and nothing else.

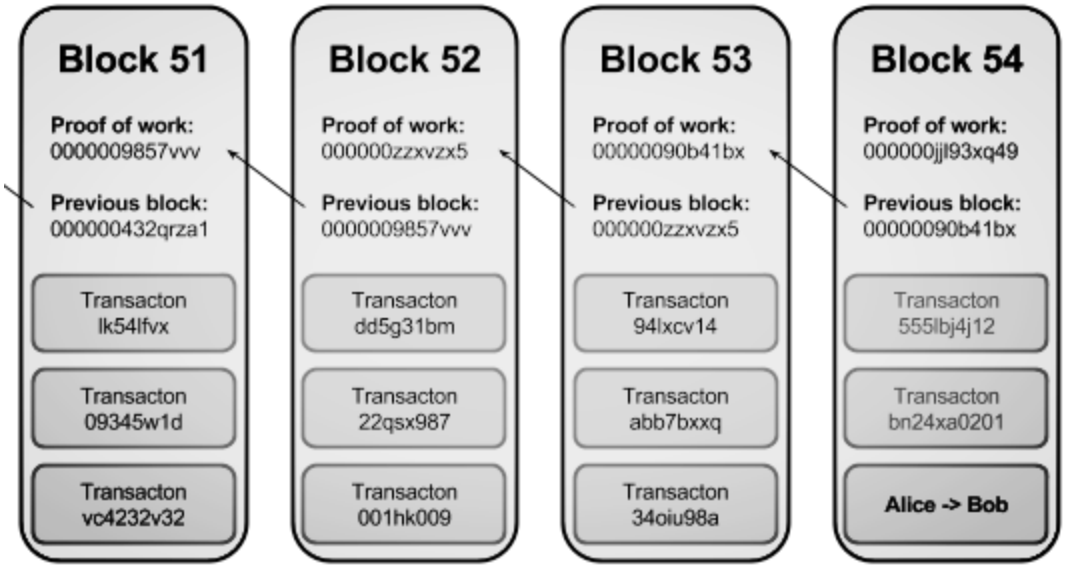


Figure 5. Blockchain data model [3]

In this case, our project aims to just work with this public information. Thus for privacy and simplification reasons we are ignoring private information, only focusing on public information-open data, which means, we are ignoring location information about nodes and addresses, IP addresses from nodes, wallet information such private keys, or any data that is not in the blockchain.

3.6 Design Constraints

We are potentially limited by the specifications of the server we are able to procure.

3.7 Logical Database Requirements

Graph database, RDF

4. Analysis Models

List all analysis models used in developing specific requirements previously given in this SRS. Each model should include an introduction and a narrative description. Furthermore, each model should be traceable the SRS's requirements.

5. Change Management Process

We will use Git as a collaborative work environment, changes will be discussed amongst the group with consultation from the mentor and a common understanding will be reached before changes are made.