ADONIS (ADS) : Blockchain, Smart Contract & Cryptocurrency System

David Levin

--------------------**

1. Introduction

i. Intro

This document presents my final project as part of the final year of a degree in Computer Science at Bar Ilan University, Israel.

ii. History

My acquaintance with the world of Cryptocurrency began exactly a year ago. I was exposed to an online article that intrigued me, and its subject was digital currencies and their future impact on the world. From that moment on I was fascinated, and decided I was starting to learn this world.

iii. About the project

As part of my project and development, I would like to explore and understand how to bring the system to as "green" a state as possible – that is, to be environmentally friendly.
Moreover, the purpose of my project is to provide a solution to work problems and business applications.

At first, I went directly to solving the problem – that is, designing the blockchain system.
It didn't take long time for me to realize that this is not the right way - and that I needed to "rethink".
I decided that in order to be able to think and plan an effective and efficient solution, I needed to deepen my knowledge on a number of different topics:

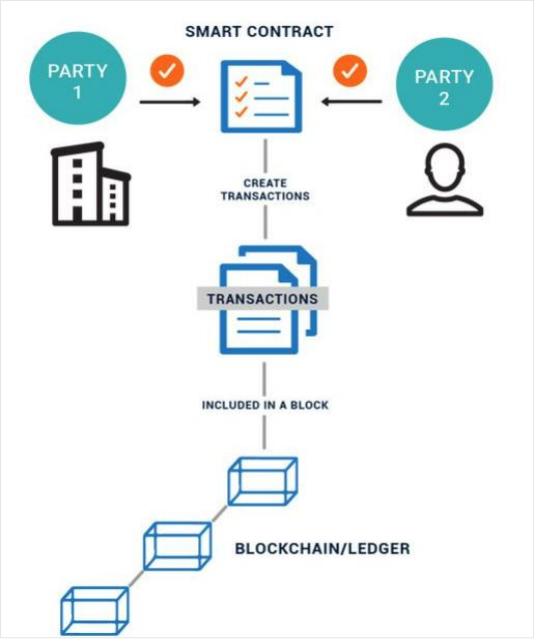
1. Blockchain system
2. Cryptocurrency
3. System of smart contracts
4. Generic trading algorithm

In order to deep my knowledge in the best way, I chose to "expand" my project a bit – and develop all 4 things I mentioned:
a blockchain system, a cryptocurrency, and a system of smart contracts – that will work perfectly together. And for "dessert", I decided to create a generic trading algorithm – one that would help generate profits.

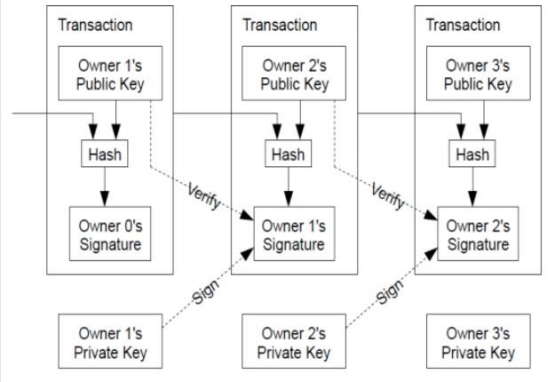
Thus, it can be said that I have now set 4 different goals for the project.

2. Methodology

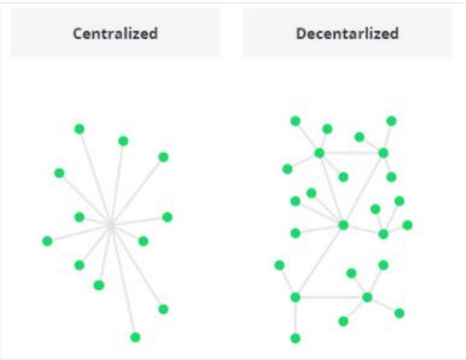
iv. Flow diagram – Blockchain and smart contracts

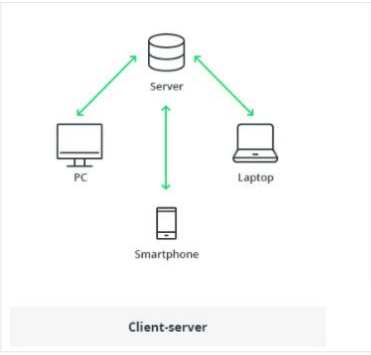


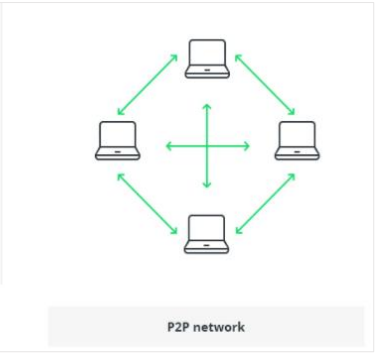
v. Blockchain – simplified view



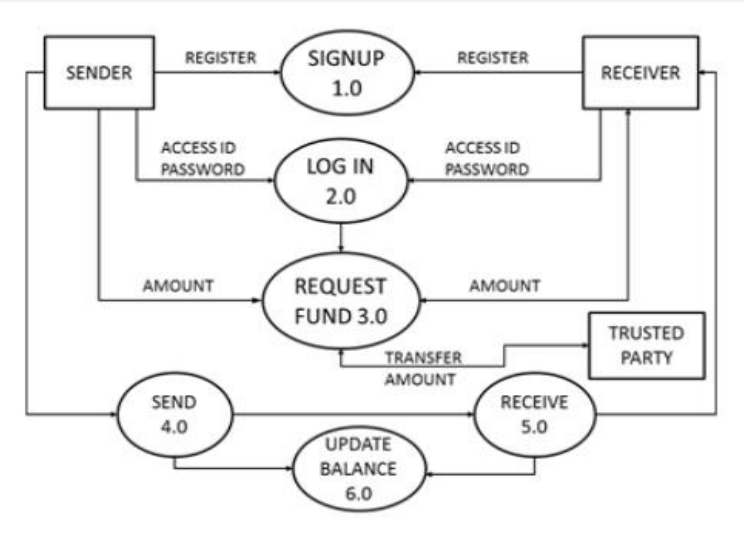
vi. Informative images:



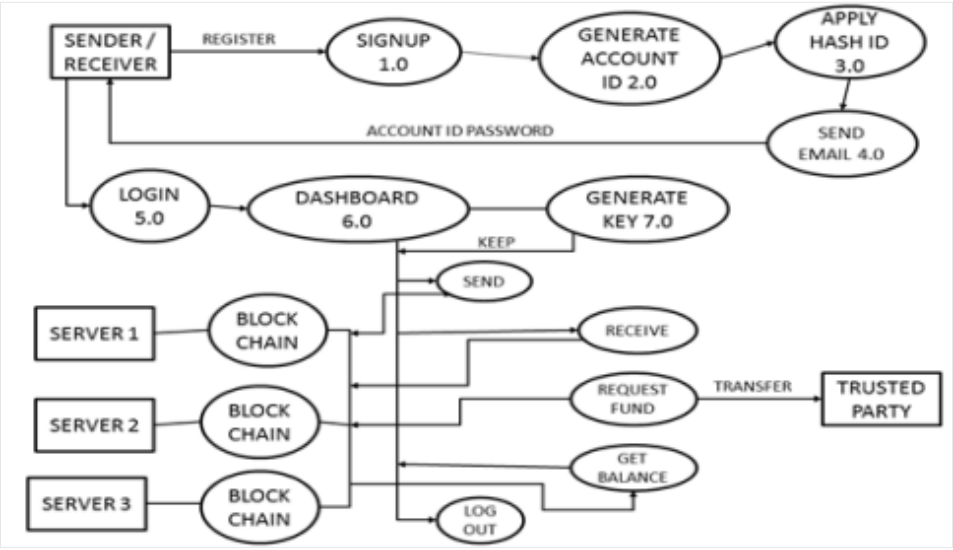




vii. User flow diagram:



viii. User flow diagram (another option):



In the [1] reference, you can see the flow diagram of the blockchain system and the smart contracts: two parties are interested in reaching some agreement regarding a particular property (where the main idea here is that there is no broker – the same banker or any other factor we are used to involving). In order to execute their contract "smartly" – we are required to involve the blockchain system.

As I mentioned earlier, this is also one of the reasons I chose to build both a Blockchain system and a smart contract system – in order to create a system that works perfectly.

In the [2] reference, you can conceptually see the Blockchain system idea: you can see how the relationship between the various person in the network is formed, in the sense that no person holds all the information about the network. In such situation, each of the partners in the network depends on its other partners – which allows the creation of a reliable and strong security mechanism.

In the [3] & [4] references, the idea of reference number [2] can be seen in a tangible way: while modern bank systems are managed in a centralized way – the whole idea in creating the blockchain network is to break this centralization, by each one depending on the other members of the network.
In this way, there is no "leader" for the network.

In the [5] & [6] references, you can see the user flow diagram of the system.
As of this point in time, I am directing the system to work in configuration of reference number 5 (the first User flow).
But there is another option (reference number 6), to which I may make some adjustments - but for now it seems less relevant (I also added this visualization, so that there will be another illustration).

3. Requirement Analysis

ix. Preliminary knowledge

Throughout the development process and in order for me to develop the product in the best way, there are a number of thinking concepts and principles that need to be exposed to them. Including:

1. Understanding SHA256 hash
2. Immutable ledger
3. Distributed P2P network
4. Byzantine fault tolerance
5. Consensus protocol
6. Decentralized .vs. Distributed
7. Mining pools
8. CPU's .vs. GPU's .vs. ASIC's
9. The 51% attack
10. The DAO attacks
11. Soft & Hard forks
12. Segregated witness
13. Public key
14. ICO case study
15. Blockchain and WEB 3.0
16. etc

x. Requirements and technological dependencies

In order to develop the entire system, I will involve different types of technologies, which will help create a strong, reliable, and durable system. The programming language I chose to use is Python. In order to best implement the smart contract system, I may choose to use C ++ instead of Python, and then I will need to build an interface that combines the two languages in the ideal way.

Furthermore, throughout the development I will use a number of development tools to help ensure that the entire system works optimally:

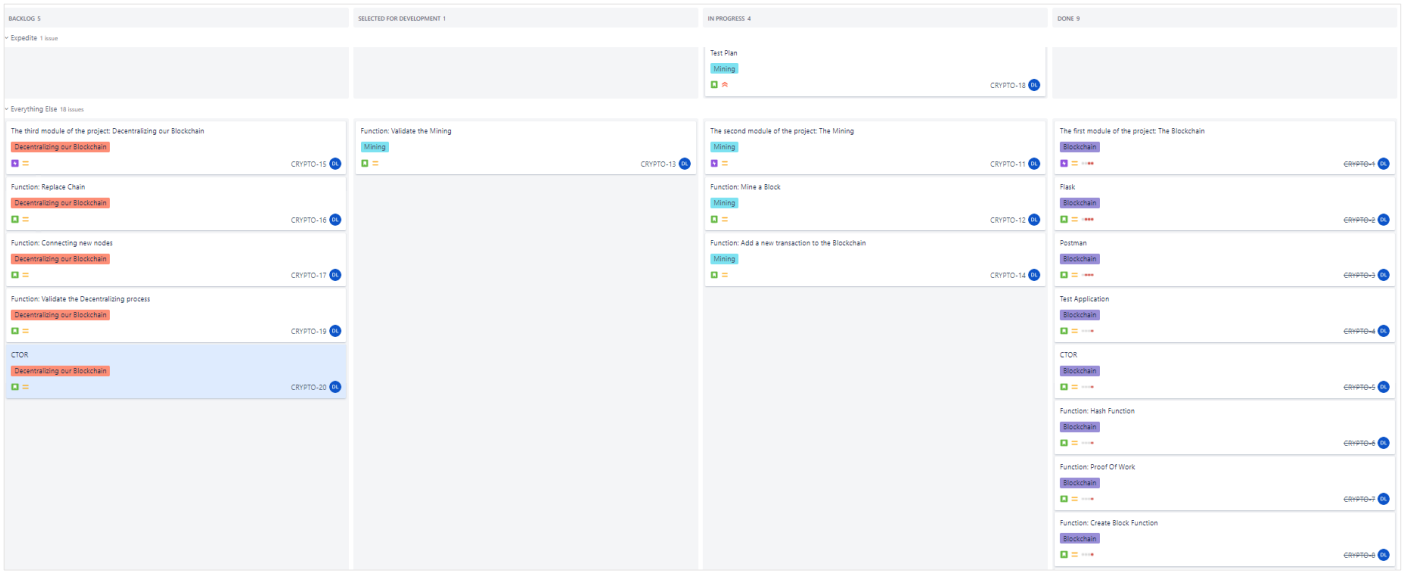
1. **Postman** – Postman is an API platform for building and using APIs. Postman simplifies each step of the API lifecycle and streamlines collaboration so you can create better APIs—faster.
2. **Flask** – a micro web framework written in Python. Flask supports extensions that can add application features as if they were implemented in Flask itself.
3. **Anaconda** – Anaconda is a distribution of Python and R programming languages for scientific programming (data science, machine learning applications, large-scale data processing, predictive analytics, etc.), which aims to simplify package management and deployment.
4. **Various scientific libraries in Python –** numpy, cv2, pandas, PyTorch etc.

4. JIRA

xi. Work control

In order for me to work in the best way, I chose to use the JIRA system for project management.
I have divided the project into a number of epics, so that for each and every one of them, I will open user stories that will explain them.
I also chose the order of work on the epics so that the system would build step by step, and that no dependencies would be created that might lead to the "crashing" of the project.

In the attached picture you can see the current board of the project in the JIRA system:
The first Epic I chose to work on is Blockchain, when I linked for it a number of user stories that will help complete the Epic goal.
It can be seen that beyond concrete tasks (such as creating functions, writing a stacking function, etc.), there are also learning tasks (for learning new technologies that I will use for the benefit of project realization).



xii. Epics

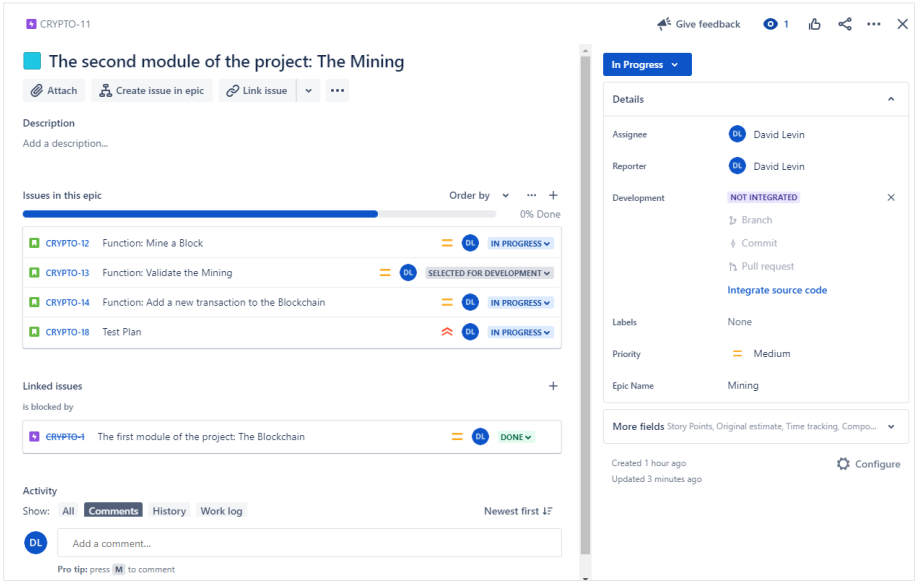
Currently, I have created 3 different epics that will best classify the work:

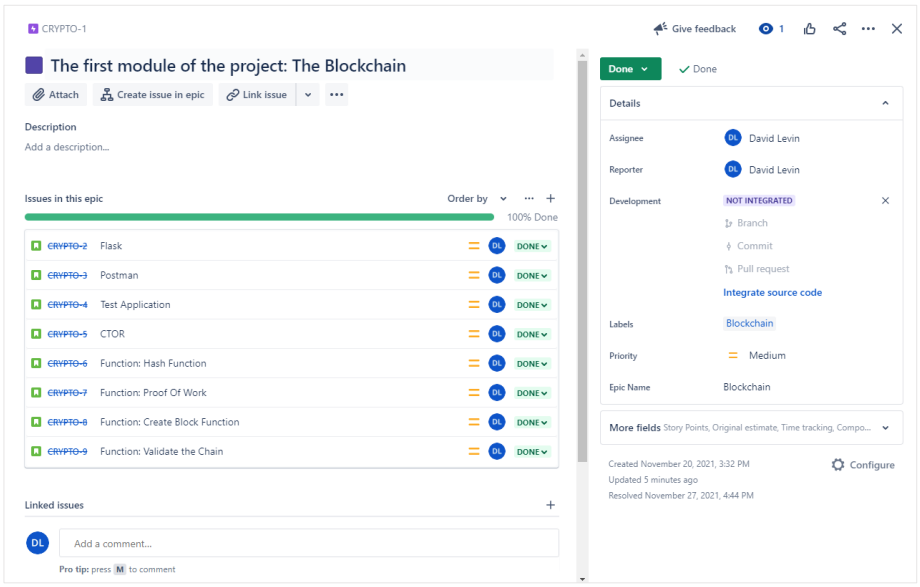
1. Blockchain (the blue epic)
2. Mining (the light blue epic)
3. Decentralizing the Blockchain (the orange epic)

As of this point in time in the project, these are the 3 epics that are needed right now. Of course, as the project progresses, I will be required to produce more epics, for example:

The website that will be used by the blockchain users, the smart contract system, the trading algorithm and more.

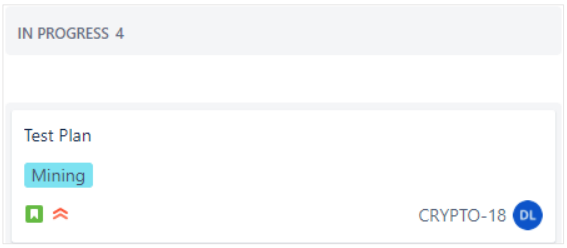
I'm adding photos for the 3 epics that I described above:





xiii. Test Plan

As you can see in the image I attached, I created a special User Story for the tests. I added the full file at the end of the above architecture file, as required.



5. Conclusion

ADONIS system overarching goal is to create a quality blockchain system - both in terms of reliability towards users, both in terms of the environment and in terms of interfacing with the system of smart contracts.
Also, create a cryptocurrency that can be traded and interfaced with the blockchain system and the smart contract system - so that all three can work accurately together.