FACULTY OF INFORMATION TECHNOLOGY CTU IN PRAGUE

Review report of a final thesis

Reviewer:	doc. Ing. Filip Křikava, Ph.D.
Student:	Bc. Lukáš Kozák
Thesis title:	Developer tooling for Solana
Branch / specialization:	Computer Systems and Networks
Created on:	31 May 2022

Evaluation criteria

1. Fulfillment of the assignment

[1] assignment fulfilled
[2] assignment fulfilled with minor objections
▸ **[3] assignment fulfilled with major objections**
[4] assignment not fulfilled

The thesis is positioned as "semi-analytical and semi-implementation". While I'm not entirely sure what does that mean (each thesis necessary needs to contain an analysis, otherwise it would not be a thesis), but it does fail on the promise that it will explain how the Solana blockchain work (cf. below).
The implementation part is described in two sentences:

1.
"The practical part of the thesis consists of implementing a new feature for Trdelnik, which is a new open-source testing framework for Solana programs based on Anchor, which is yet a new development framework that helps to simplify the process of building new and secure Solana on-chain programs."

2.
"Because writing raw Solana programs is a tedious and error-prone process, Anchor, together with Trdelnik, aim to fill most of the gaps in the current development ecosystem of Solana and bring better developer experience to everyone."

That is hard to parse, but most importantly, nowhere at the beginning it is said what is this "new feature" and most importantly, why is it important.

2. Main written part

60 / 100 (D)

The first two paragraphs of the introduction are promising. Entering the world of cryptocurrency and blockchains, they give an impression that in the thesis one will learn

about a fairly new blockchain Solana with its new inventions which aim to progress in solving the scalability trilemma.
Unfortunately, right after that it falls down the road.
First: in the goal of the thesis there is a claim that "there is little to no officially-provided correct information, and currently, it is not easy to understand Solana from the ground up without digging into the source code or personally asking the developers how some of the specifics work." but with no specifics. Does this mean that the official documentation is incorrect? What exactly is missing or wrong? How can people use the network if there are no information or are incorrect?
After reading the next 24 pages dedicated to Solana, I do not think I understand it any better than before. Finally, I had to reach to the for Yakovenko's Solana's white paper to learn how the essential pieces such as Proof-of-History and the actual consensus work in this network.
The problem is that there is basically no story, only a large number of terms that are described into various degree of details, but no synthesis is ever presented. This is a pity as there is a clear story.
I understand that the world of cryptocurrencies is confusing and extremely fast moving. I was hoping that the thesis will explain some essential parts of how the network work, how does it differ from the main competitors such as Ethereum (after the London update) or Cardano?
How dapps differ from the normal apps? What different developer tools do you need? How does the Rust programming model provided by Solana differs from other languages / models used on blockchains such as Plautus, Marlow, Dime, ...
what would seriously help is some running example on which many of the concepts could be explained instead of dry enumeration of terms.

Moving onto the implementation description. Again IN section 2.2 there is a an enumeration of FR and NFR without any discussion why. Finally, the chapter about assessment really feels like it was sewn with a hot needle. Everything apart, how can a correctness of an implementation be verified by unit testing? High test coverage is good, but what can it say about correctness?

Few details:

p8: missing citations. If I'm not mistaken "In other words, with synchronized clocks, we can replace communication with local computation." was said by B. Liskov in [2]
p9: should not be the upper bound: Message_i must have taken place *before* Hash_{i+1}?
p10: "The exact description is out of the scope of this work." A brief description will help - or at least remove the figure of you do not talk about it.
p15: "Sealevel hands off transactions to be executed on hardware natively using an industry-proven bytecode called the Berkeley Packet Filter (BPF)" - it can be either a native code which runs natively or a byte code which runs in a VM
p17: Not sure how to read the Table 1.1 - either discuss more or remove
p18: What is the point of the two listings? Why are they relevant? This page promises to explain the programming model, but again it is just a sequence of terms.
p26: Project Serum, Bonfida DEX - are we suppose to know these? Why to iterate names without any links or description?
p28: The point 3 is confusing: what does that mean - the coherency of the network can be compromised by a malicious app?
p31: Listing 1.4 onwards - is this your code?

p34: "A safe language like Rust should help minimize issues that would be left undiscovered using any other language for testing." - which issues? What about Plautus?
p41: Missing relationships with FR
p43: "Thanks to the design of the language, the mistakes and common undefined behavior known from languages, such as C or C++ are caught during compilation time." Which mistakes?
p43: "Rust's zero-cost abstractions result in comparable performance to code written in C or C++ and are
slowly becoming their contender even in the field of High-Performance Computing." - claims, missing citations
p43: "As not many are familiar with Rust, to not get lost in it terminology, let some of the key terms of a typical Rust project be repeated." - this is a thesis not a blog :-)
p44: Rust crates are organized into a package not a workspace AFAIK
p49: Why is it interesting to see a listing of a HashMap initialization with values and keys?

---

[1] A. Yakovenko, Solana: A new architecture for a high performance blockchain v0.8.13
[2] B. Liskov, Practical Uses of Synchronized Clocks in Distributed Systems, 1991

3. Non-written part, attachments 75 /100 (C)

The link to to github repo / the SD card contains a code of a project with 6 contributors. I guess the code which represents this thesis is in the "crates/explorer". There is about 4.5K lines of Rust code. The code looks fine. Since there is close to no description in the thesis I'm not sure what am I looking for. I managed to run the tests.

4. Evaluation of results, publication outputs and awards 70 /100 (C)

Not sure how to assess this - there is implementation merged into project that seems to be part of some service offering thus I assume it does what it should.

The overall evaluation 60 /100 (D)

I'm disappointed by this thesis. I cannot well judge the implementation side, but the written part is below what I would expect from a master thesis. It fails to deliver its promise and pretty much it contains a disjoint blobs of text.

Questions for the defense

Since it was not in the thesis and I'm still curious:

1. How does Solana differ from Cardano and Ethereum London?
2. What are the differences between the programming model provided by the Rust API for Solana and say the programming languages for Cardano or Ethereum?
3. More open-ended, I'm wondering why does Solano suffer from outages so often (already 7 times in 2022)? Is it bad luck (in the sense of attracting more attacks) or are there some architecture-level problems?

Instructions

Fulfillment of the assignment

Assess whether the submitted FT defines the objectives sufficiently and in line with the assignment; whether the objectives are formulated correctly and fulfilled sufficiently. In the comment, specify the points of the assignment that have not been met, assess the severity, impact, and, if appropriate, also the cause of the deficiencies. If the assignment differs substantially from the standards for the FT or if the student has developed the FT beyond the assignment, describe the way it got reflected on the quality of the assignment's fulfilment and the way it affected your final evaluation.

Main written part

Evaluate whether the extent of the FT is adequate to its content and scope: are all the parts of the FT contentful and necessary? Next, consider whether the submitted FT is actually correct – are there factual errors or inaccuracies?

Evaluate the logical structure of the FT, the thematic flow between chapters and whether the text is comprehensible to the reader. Assess whether the formal notations in the FT are used correctly. Assess the typographic and language aspects of the FT, follow the Dean's Directive No. 52/2021, Art. 3.

Evaluate whether the relevant sources are properly used, quoted and cited. Verify that all quotes are properly distinguished from the results achieved in the FT, thus, that the citation ethics has not been violated and that the citations are complete and in accordance with citation practices and standards. Finally, evaluate whether the software and other copyrighted works have been used in accordance with their license terms.

Non-written part, attachments

Depending on the nature of the FT, comment on the non-written part of the thesis. For example: SW work – the overall quality of the program. Is the technology used (from the development to deployment) suitable and adequate? HW – functional sample. Evaluate the technology and tools used. Research and experimental work – repeatability of the experiment.

Evaluation of results, publication outputs and awards

Depending on the nature of the thesis, estimate whether the thesis results could be deployed in practice; alternatively, evaluate whether the results of the FT extend the already published/known results or whether they bring in completely new findings.

The overall evaluation

Summarize which of the aspects of the FT affected your grading process the most. The overall grade does not need to be an arithmetic mean (or other value) calculated from the evaluation in the previous criteria. Generally, a well-fulfilled assignment is assessed by grade A.