Marvin Siu Yin Chan

Revised Project Proposal

This is a brand new project proposal.

The invention of the blockchain has allowed for a new computing paradigm - a programmable computer that can be used by anyone but is not owned by anyone - and code deployed on the blockchain is called a smart contract. This new computing paradigm has changed industries like finance and art in the form of decentralized finance and non-fungible tokens, respectively. However, since anyone can interact with any smart contract on the blockchain, security vulnerabilities in smart contracts can be exploited by malicious actors to generate undesirable transactions. Recent hacks on Axie Infinity and Wormhole have resulted in $600 million and $325 million worth of assets stolen, respectively. These hacks highlight the importance of security in smart contracts.

There are currently many blockchains like Ethereum and Solana, that support smart contracts, and the largest of them all is Ethereum. The most popular language used to develop Ethereum smart contracts is Solidity with 8,000+ public repositories on GitHub. Solidity is a JavaScript-like, Turing-complete language and has been in constant development since 2014 by the Ethereum team. Solidity gives developers a lot of flexibility, but this flexibility has also resulted in many security flaws. As a result, developers were looking to develop a new language that limits security flaws and came up with Vyper. Vyper is a pythonic programming language that is non-Turing complete. As a design choice, Vyper has also removed many features in Solidity to decrease security risks like inline assembly. Vyper is much newer than Solidity, and the first stable version of Vyper was released in 2020. Hence, there are only 84 public Vyper repositories on GitHub.

For developers that want to learn smart contract development, one has to choose whether to learn Solidity or Vyper. Vyper's focus on security, language simplicity and auditability make it more appealing, but the newness of the language means that there is limited community and support. For my project, the primary research question I want to answer is: how difficult it is to learn and develop a smart contract in Vyper for experienced developers new to smart contract development.

To gauge how difficult it is to learn and develop a smart contract in Vyper, I plan to reimplement the Uniswap V2 smart contract. Uniswap V2 smart contract, which is written in Solidity, facilitates swapping between cryptocurrencies and it is the 3rd most widely used smart contract on Ethereum. I chose to reimplement an existing Solidity smart contract because the features are well defined making it easier to test correctness. While I am reimplementing the smart contract, I will keep track of the amount of time spent, the number of google searches, the resources used, etc.

I understand that the results of the project are dependent on prior development experience and knowledge of blockchain, and one could argue that the results would only be useful to me. However, I would say my background matches the background of many people trying to get into the blockchain industry. I have worked a couple of years as a Data Engineer and am about to complete an MSCS. Hence, I would generalize myself as an experienced developer. I have also been a user of the blockchain since 2021, a year when Metamask saw monthly active users grow from 500,000 to 10,000,000. As a result, I would say my profile fits many of the people that are thinking of going into smart contract development, and this project would be of value to them.

Wormhole Hack:
https://www.theverge.com/2022/2/3/22916111/wormhole-hack-github-error-325-million-theft-ethereum-solana
Axie Infinity Hack:
https://www.theverge.com/2022/3/29/23001620/sky-mavis-axie-infinity-ronin-blockchain-validation-defi-hack-nft
Uniswap v2: https://github.com/Uniswap/v2-core
Solidity: https://docs.soliditylang.org/en/v0.8.13/
Vyper: https://vyper.readthedocs.io/en/stable/index.html