

COMMITTABLE: A Decentralised and Trustless Open-Source Protocol

Han Liu

University College Oxford
Blockchain Research Centre
Oxford, UK
liuhan@committable.io

Huafeng Zhang

University College Oxford
Blockchain Research Centre
Oxford, UK
huafeng@committable.io

Bangdao Chen

University College Oxford
Blockchain Research Centre
Oxford, UK
bangdao.chen@gmail.com

A.W. Roscoe

University of Oxford
TBTL
University College Oxford
Blockchain Research Centre
Oxford, UK
awroscoe@gmail.com

Abstract—Collaborative development in open-source software (OSS) has long been limited by the lack of participation, *i.e.*, A project is often maintained by an insufficient number of developers, especially for small- and medium-size projects. To establish a sustainable ecosystem for global developers and projects, we propose a decentralised and trustless open-source protocol **Committable** for all OSS software. The key insight behind **Committable** is an accountable and trusted tokenisation technology on blockchain that creates the CMT software assets for a variety of artefacts (*e.g.*, document, code, testcase, makefile *etc.*) across the whole development lifecycle. A CMT token defines an abstraction of commits to OSS and systematically models the contribution from developers, therefore is far more comprehensive than a commit hash that are commonly used to identify software versions. In further, **Committable** introduces the **Problem-Solution-Risk** (PSR) framework to evaluate and reward a given set of CMT in an unbiased manner based on their contributions to a project. Owners of CMT are allowed to trade their tokens in the marketplace on blockchain established by **Committable**. The trading of CMT leads to transfer of token rights (*e.g.*, sell, earn PSR rewards *etc.*), and more importantly, royalty to the developer for his or her original contribution. This demonstration proposal will introduce **Committable** on the test net of Ethereum and describe a preliminary case study with the **OpenZeppelin** project.

Index Terms—open-source software, blockchain, tokenisation

I. INTRODUCTION

The open-source software (OSS) has been undergoing a rapid growth in recent years. As a form of practice for common good, OSS is collaboratively developed and maintained by a decentralised group of developers, who are potentially from different organisations or as individual developers. While a very limited number of OSS projects are well supported by big enterprises, *e.g.*, Google, Facebook, the majority of OSS projects are unfortunately having a hard time to sustain participating developers. Based on the 2020 GitHub insight report [1], over 99% of OSS projects are inactive and more than 70% of them are managed by less than 10 developers. As many OSS projects are playing a fundamental role in nowadays digital systems, *e.g.*, OpenSSL for web services, the lack of sustainability threatens the entire OSS ecosystem in the long run.

The **Committable** project is particularly designed to establish an open and sustainable ecosystem for all OSS. As a decentralised and trustless application, **Committable** requires no trusted third parties and can be instantiated with any existing blockchain that supports smart contracts. The key capability of it comes from an accountable and trustless tokenisation technology¹ which monetises commits to software artefacts (*e.g.*, document, code, testcase, makefile *etc.*) as non-fungible tokens on blockchain, *i.e.*, CMT in the case of **Committable**. The tokenisation is achieved through well-defined software analysis (*e.g.*, abstract-syntax tree parsing, static source code and binary analysis *etc.*) to abstract the development process in a way that structural and semantic updates are considered instead of textual modifications. Moreover, all related parties would be able to locally verify CMT and its tokenisation process as well. In general, **Committable** enables three main functionalities with the CMT token as below.

- **Tokenisation.** Developers can mint CMT tokens from their commits on accepted platforms, *e.g.*, GitHub [2]. Each commit maps to at most one unique token and the tokenisation is only valid when requested by the original developer.
- **Trade.** The marketplace service is provided for CMT owners to trade their tokens at a specified price. The trading transfers the ownership from the seller to the buyer and involves a royalty rewarded to the account which minted the CMT in the first place.
- **Payment.** An OSS project owner can set up a payroll pool for a collection of CMT and pay to them based on a fair evaluation that decides their contributions to the host project.

II. DESIGN OF **Committable**

The architecture of **Committable** is shown in Figure 1. The demo is on Ethereum [3] and connects to GitHub [2].

Trusted Services. The application interacts with users through a web client where a user can choose to login via GitHub and MetaMask [4] to have a full support from **Committable**, *e.g.*, mint and trade tokens *etc.*. One of the key components in **Committable** is a set of services that run in a

¹This technology is currently in a process of patent application.

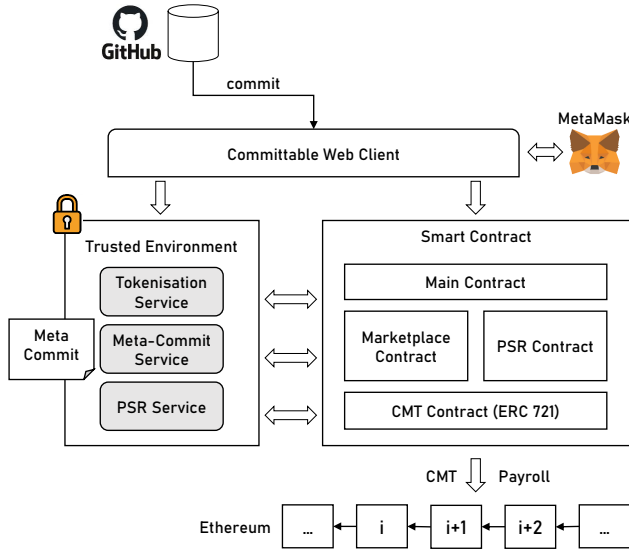


Fig. 1: The architectural design of Committable.

trusted environment, as displayed in the left part of Figure 1. In our implementation, the environment is achieved by configuring a trusted cloud sever with the support of trusted execution environment (TEE), *i.e.*, AMD SEV [5] to be specific. The environment certifies that the services are indeed running inside such an environment with implementations that are cryptographically the same as claimed. Inside the cloud server, Committable sets up three trusted services. Specifically, the *Tokenisation* service monetises a given GitHub commit as the CMT token via clearly-defined software analysis. Such an analysis commonly explores the a representation of the target software both automatically and systematically to generate a computational abstraction on the fly. In addition to the minted CMT, the tokenisation generates a piece of meta commit data which describes core attributes of the abstraction. The meta-commit is managed by the *Meta-Commit* service in a secure way since anyone can run the tokenisation locally to check whether the local meta-commit is in line with the remote version. The *PSR* service, *i.e.*, short for *Problem-Solution-Risk*, algorithmically evaluates a given set of CMT with a vector of “scores” based on their contributions to the project. The principle of PSR is to highlight CMT which addresses prioritised problems with effective solutions and a limited level of risks. More importantly, the trusted environment can be openly configured in the ecosystem, *i.e.*, anyone can submit his or her implementations to those services.

Smart Contracts. In the right part of Figure 1, Committable creates a set of smart contracts to instantiate a decentralised protocol with key functionalities as aforementioned. The CMT token implements the ERC-721 interface [6] as a non-fungible token on Ethereum. Based on the CMT contract, Committable defines the *Main* contract to manage the processing of incoming transactions. The requests for trading are handled by the *Marketplace* contract via making orders between buyers and

sellers. Moreover, the *PSR* contract allows an OSS owner to create a money pool on blockchain with accepted types of cryptocurrencies, *e.g.*, ETH, USDT *etc.*, and further distribute the pool to relevant CMT tokens based on a payroll decided by the evaluation from the *PSR* service. The ownership of CMT and decision of payroll are stored on the blockchain and become consensus for all participants.

III. APPLICATION AND DEMONSTRATION PLAN

Figure 2 shows screenshots of Committable. Specifically, Figure 2a outlines the CMT owned by a user, each of which points to the storage of meta-commit. Figure 2b demonstrates the page of marketplace to buy or sell CMT tokens. Figure 2c and 2d are the interfaces to create a money pool and check payments from the payroll.

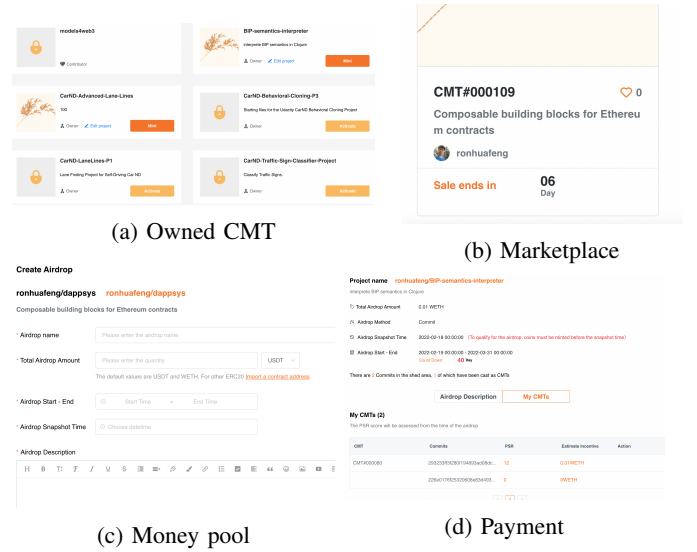


Fig. 2: Screenshots of the Committable application

In the case study with the OpenZeppelin project [7], Committable maps to human opinions with a 0.7 Pearson Correlation Coefficient in terms of CMT evaluation.

Demo Plan. We plan to walk through the Committable application with a live demo on a selected OSS project. The demonstration starts from a login via GitHub and MetaMask. Then, we mint CMT tokens from commit records and further explain the meta-commit associated with the token. We also plan to try out trading with CMT in the marketplace and deciding a fair payroll for a specific set of CMT tokens.

REFERENCES

- [1] “2020 github insight report,” <http://oss.x-lab.info/github-insight-report-2020-en.pdf>, 2020.
- [2] “Metamask,” <https://metamask.io/>, 2022.
- [3] G. Wood *et al.*, “Ethereum: A secure decentralised generalised transaction ledger,” *Ethereum project yellow paper*, vol. 151, no. 2014, pp. 1–32, 2014.
- [4] “Github,” <https://github.com/>, 2022.
- [5] “Amd secure encrypted virtualization,” <https://developer.amd.com/sev/>, 2022.
- [6] “Erc 721 standard,” <https://eips.ethereum.org/EIPS/eip-721>, 2022.
- [7] “Openzeppelin contracts,” <https://github.com/OpenZeppelin/openzeppelin-contracts>, 2022.