CertiK white paper

draft for open community review and subject to change

CertiK: Building Fully Trustworthy Smart Contracts and Blockchain Ecosystems

www.certik.org

December 4, 2017 to March 27, 2019

IMPORTANT NOTICE

NOTHING IN THIS WHITEPAPER CONSTITUTES LEGAL, FINANCIAL, BUSINESS OR TAX ADVICE AND YOU SHOULD CONSULT YOUR OWN LEGAL, FINANCIAL, TAX OR OTHER PROFESSIONAL ADVISER BEFORE ENGAGING IN ANY ACTIVITY IN CONNECTION HEREWITH. NEITHER CERTIK FOUNDATION LTD. (THE FOUNDATION), ANY OF THE PROJECT TEAM MEMBERS WHO HAVE WORKED ON THE CERTIK SOFTWARE (AS DEFINED HEREIN) OR PROJECT TO DEVELOP THE CERTIK SOFTWARE IN ANY WAY WHATSOEVER (THE CERTIK TEAM), ANY DISTRIBUTOR/VENDOR OF CKT AND CKG (THE DISTRIBUTOR), NOR ANY SERVICE PROVIDER SHALL BE LIABLE FOR ANY KIND OF DIRECT OR INDIRECT DAMAGE OR LOSS WHATSOEVER WHICH YOU MAY SUFFER IN CONNECTION WITH ACCESSING THIS WHITEPAPER, THE CERTIK FOUNDATION WEBSITE AT HTTPS://CERTIK.ORG OR ANY OTHER WEBSITES OR MATERIALS PUBLISHED BY THE FOUNDATION.

PLEASE NOTE: CRYPTOGRAPHIC COINS REFERRED TO IN THIS WHITE PAPER REFER TO CRYPTOGRAPHIC COINS ON A LAUNCHED BLOCKCHAIN THAT ADOPTS THE CERTIK SOFTWARE. THEY DO NOT REFER TO THE ERC-20 COMPATIBLE TOKENS OR ANY OTHER TOKENS BEING DISTRIBUTED ON SOME BLOCKCHAINS IN CONNECTION WITH THE CKT/CKG DISTRIBUTION.

This Whitepaper is intended for general informational purposes only and does not constitute a prospectus, an offer document, an offer of securities, a solicitation for investment, or any offer to sell any product, item or asset (whether digital or otherwise). The information herein below may not be exhaustive and does not imply any elements of a contractual relationship. There is no assurance as to the accuracy or completeness of such information and no representation, warranty or undertaking is or purported to be provided as to the accuracy or completeness of such information. Where this Whitepaper includes information that has been obtained from third party sources, the Foundation and/or the CertiK team have not independently verified the accuracy or completion of such information. Further, you acknowledge that circumstances may change and that this Whitepaper may become outdated as a result; and the Foundation is under no obligation to update or correct this document in connection therewith.

This Whitepaper does not constitute any offer by the Foundation, the Distributor or the CertiK team to sell any CKT and CKG (as defined herein) nor shall it or any part of it nor the fact of its presentation form the basis of, or be relied upon in connection with, any contract or investment decision. Nothing contained in this Whitepaper is or may be relied upon as a promise, representation or undertaking as to the future performance of the CertiK Software. The agreement between the Distributor and you, in relation to any sale and purchase of CKT and CKG is to be governed by only the separate terms and conditions of such agreement.

By accessing this Whitepaper or any part thereof, you represent and warrant to the

Foundation, its affiliates, and the CertiK team as follows:

(a) you acknowledge, understand and agree that CKT and CKG may have no value, there is no guarantee or representation of value or liquidity for CKT/CKG, and CKT/CKG is not for speculative investment;

(b) none of the Foundation, its affiliates, and/or the CertiK team members shall be responsible for or liable for the value of CKT/CKG, the transferability and/or liquidity of CKT/CKG and/or the availability of any market for CKT/CKG through third parties or otherwise;

(c) in any decision to purchase any CKT/CKG, you have not relied on any statement set out in this Whitepaper;

(d) you will and shall at your own expense ensure compliance with all laws, regulatory requirements and restrictions applicable to you (as the case may be); and

(e) you acknowledge, understand and agree that you are not eligible to purchase any CKT/CKG if you are a citizen, national, resident (tax or otherwise), domiciliary and/or green card holder of a geographic area or country (i) where it is likely that the sale of CKT/CKG would be construed as the sale of a security (howsoever named) or investment product and/or (ii) in which access to or participation in the CKT/CKG sale or the CertiK Software is prohibited by applicable law, decree, regulation, treaty, or administrative act, and/or (including without limitation the United States of America, Canada, New Zealand, People's Republic of China and the Republic of Korea).

The Foundation, the Distributor and the CertiK team do not and do not purport to make, and hereby disclaims, all representations, warranties or undertaking to any entity or person (including without limitation warranties as to the accuracy, completeness, timeliness or reliability of the contents of this Whitepaper or any other materials published by the Foundation). To the maximum extent permitted by law, the Foundation, the Distributor, their related entities and service providers shall not be liable for any indirect, special, incidental, consequential or other losses of any kind, in tort, contract or otherwise (including, without limitation, any liability arising from default or negligence on the part of any of them, or any loss of revenue, income or profits, and loss of use or data) arising from the use of this Whitepaper or any other materials published, or its contents or otherwise arising in connection with the same. Prospective purchasers of CKT/CKG should carefully consider and evaluate all risks and uncertainties (including financial and legal risks and uncertainties) associated with the CKT/CKG sale, the Foundation, the Distributor and the CertiK team.

All contributions will be applied towards the Foundation's objects, including without limitation promoting the research, design and development of, and advocacy for a decentralised community system which would promote the establishment of a formal verification framework for building fully reliable, secure and hacker-resistant smart contacts and blockchain ecosystems.

The information set out in this Whitepaper is for community discussion only and is not legally binding. The agreement for sale and purchase of CKT/CKG and/or continued holding of CKT/CKG shall be governed by a separate set of Terms and Conditions or CKT/CKG Purchase Agreement (as the case may be) setting out the terms of such purchase and/or continued holding of CKT/CKG (the Terms and Conditions), which shall be separately provided to you or made available at https://certik.org. In the event of any inconsistencies between the Terms and Conditions and this Whitepaper, the Terms and Conditions shall prevail.

No regulatory authority has examined or approved of any of the information set out in this Whitepaper. No such action has been or will be taken under the laws, regulatory requirements or rules of any jurisdiction. The publication, distribution or dissemination of this Whitepaper does not imply that the applicable laws, regulatory requirements or rules have been complied with.

All statements contained in this Whitepaper, statements made in press releases or in any place accessible by the public and oral statements that may be made by the Foundation, the Distributor and/or the CertiK team may constitute forward-looking statements (including statements regarding intent, belief or current expectations with respect to market conditions, business strategy and plans, financial condition, specific provisions and risk management practices). You are cautioned not to place undue reliance on these forward-looking statements given that these statements involve known and unknown risks, uncertainties and other factors that may cause the actual future results to be materially different from that described by such forward-looking statements, and no independent third party has reviewed the reasonableness of any such statements or assumptions. These forward-looking statements are applicable only as of the date of this Whitepaper and the Foundation and the CertiK team expressly disclaims any responsibility (whether express or implied) to release any revisions to these forward-looking statements to reflect events after such date.

This Whitepaper may be translated into a language other than English and in the event of conflict or ambiguity between the English language version and translated versions of this Whitepaper, the English language version shall prevail. You acknowledge that you have read and understood the English language version of this Whitepaper.

No part of this Whitepaper is to be copied, reproduced, distributed or disseminated in any way without the prior written consent of the Foundation.

CertiK: Building Fully Trustworthy Smart Contracts and Blockchain Ecosystems

Draft for open community review and subject to change

The CertiK Software is envisaged to be a formal verification framework for building fully trustworthy smart contracts and blockchain ecosystems. While traditional testing approaches merely attempt to detect bugs, the CertiK Software provides mathematical proofs that blockchain ecosystems are bug-free. The Foundation has developed modular verification techniques to decompose such an otherwise prohibitive proof task into smaller ones that can be solved and automatically validated in a decentralized style. These proof objects can be built and encoded in the CertiK Software and will then be validated by other participants. The certificates generated by the CertiK Software guarantee the end-to-end correctness and security of the verified smart contracts, libraries of decentralized applications (DApp), and the implementations of virtual machines/blockchains. That is also why these are called certified blockchain ecosystems.

The CertiK team comprises world-class formal verification experts who are professors from Columbia University and Yale University, as well as senior software engineers and researchers from Google, Facebook, and Microsoft Research. CertiK's founders were internationally recognized for building the world's first fully verified concurrent OS kernel, CertiKOS, using the Coq proof assistant. CertiKOS is a core component of an NSF "Expeditions in Computing" project [DeepSpec], which was nominated and selected as research highlights of CACM, and has been widely considered "a real breakthrough" toward hacker-resistant systems [YaleNews, IBTimes, YDN]. These previous successes indicate that CertiK's techniques will revolutionize blockchain ecosystems and help key ecosystem players establish higher standards of true reliability and security.

Contents

1 Introduction

Blockchain technologies, pioneered by Bitcoin [5] and Ethereum [6], provide globally-consistent ledgers that do not rely on central trusted authorities. These ledgers can record the transactions of virtual currencies by a collaboration of network nodes. The Proof-of-Work (POW) [5] or Proof-of-Stake (POS) [16] mining schemes set up a theoretically unaffordable computational cost to protect the integrity of transactions and address the double-spending problem. Therefore, it seems that the ledgers are "trustable" even without a central authority. Based on this trust, smart contracts [17] and other forms of decentralized applications (DApp) can be stored/executed in the ledgers that form the blockchain ecosystems, whose source code is entirely "transparent" to the public. In these ecosystems, the central authority is replaced by a shared consensus among network nodes, and the value is created through trust.

However, blockchain ecosystems are not truly trustworthy. Due to the transparent nature of the source code and the potentially lucrative benefits that one may receive from conducting a successful attack, these ecosystems are, in reality, highly sensitive to attacks, and far more vulnerable than expected:

- Although the protocol is well-designed and highly reliable, the ledger implementations may have flaws, as is common in other complex systems, that do not fully meet the intentions of the protocol (or specification). For example, there are 703 open issues and 2,186 closed issues reported for the official Ethereum Virtual Machine (EVM) implementation on Jan 15, 2018 [15]. Some of these issues may very well compromise the guarantees of blockchain ecosystems from the root.

- The implementation of cryptographic software libraries are also error-prone [18]. These bugs can expose security risks that may allow the digital signature mechanism to be bypassed, leading to huge financial losses.

- The open sourced and transparent nature of smart contracts and other DApps exposes the entirety of their design, as well as the underlying source code, to everyone, including malicious users. These features make some applications, like digital wallets, attractive, but also defenseless against hackers. For instance, a variant of a well-documented reentrancy attack was notoriously exploited in TheDAO [19] digital wallet, leading to the theft of more than $50M worth of Ether.

- Since blockchain ecosystems are decentralized (or unsupervised), any actions must reach consensus among the majority of network nodes in order to become effective. Thus, once the DApps are released, it may become difficult and sometimes impossible to fix bugs. Take TheDAO attack again as an example. Recovering TheDAO funds required a hard fork of the blockchain. This poses a new requirement for DApps – they have to be truly trustworthy before being uploaded to the ledgers.

State-of-the-art approaches There are many ways to improve the reliability and security of software systems, but none of them can fully address these challenges introduced by blockchains. Testing is currently the most widely used approach to enhance the trust of systems. However, as Dijkstra said, program testing can be used to show the presence of bugs, but never to show their absence [8]. Processes aimed at showing the presence of bugs will only uncover a subset of existing bugs, whereas the approach of showing the absence of bugs will uncover all possible bugs. It is obvious that using testing alone cannot eliminate the zero-day vulnerability issues.

Formal verification is an alternative approach that aims to mathematically prove that the system is correct with respect to its specifications. However, it is still difficult to formally verify practical and complex systems. Traditional verification techniques, like model checking [11], are limited to ensuring functional correctness, and suffer from the state explosion problem [12] when dealing with concurrent/decentralized programs. For that reason, some researchers [13] insist on developing mechanized proofs for functional correctness using proof assistants. This approach enables the handling of richer properties but requires substantial proof efforts. In fact, while such "formal verification" concept dates back to the 1960s, complete formal proofs of non-trivial sequential systems only became feasible recently, as demonstrated by seL4 in 2009 [7]. This result was encouraging, and it seemed that researchers were close to building an entirely verified concurrent/decentralized practical system using reasonable proof efforts. However, nine years had passed and this last step wass still insurmountable. In the single-core setting, the cost of such verification is already prohibitive; seL4 took 11 person-years to verify 7,000 lines of C code. Several researchers [9, 10] believe that it is impossible to fully verify practical concurrent/decentralized systems like blockchain ecosystems, and even if it can be done, the costs would far exceed the costs of sequential systems. To address these challenges, one will have to answer the following questions:

- *What to prove?* Most of the existing verification services can only prove that the program satisfies a list properties, e.g., "no stack overflow", "all exceptions have been handled", etc. However, such a list of properties is insufficient for ensuring that the program implements the functionality correctly. Instead, the functional correctness of these programs would need to be proven. However, writing down the functional specifications alone is a complicated affair. It requires a deep understanding of the entire system and a rigorous method of expressing the desired system behaviors.

- *How to scale the proof development?* The cost of current proofs has become a significant obstacle. There is a need to further cut down the proof efforts, making it possible for one project to borrow intelligence and computation resources from a broader community.

- *How to let others trust the proofs?* Developing a proof method is hard, but it is even harder to convince people that the method is sound. It is not very meaningful to force others to trust some so-called "black-box" proofs without understanding how and why these proofs work. There is a need to allow people to validate proofs on their own local machines and encourage others to participate in this validation procedure.

1.1 CertiK Software Overview

It is believed that the answers to the above questions are rooted in the blockchain itself. This belief has inspired the Foundation to develop a one-stop solution, named the CertiK Software, which provides a powerful set of Certified Kits for building fully trustworthy blockchain ecosystems, including the CertiK certified blockchain, the CertiK verification platform, and the CertiK dual-coin model that powers the entire CertiK ecosystem.

CertiK Blockchain

The blockchain implementation serves as the backbone of the computing layer of blockchain ecosystems. A single bug in its implementation can make the entire ecosystem vulnerable to attacks. Instead of continuously detecting and fixing bugs in the codebase of existing blockchains, CertiK proposes a constructive approach: building a certified blockchain from scratch. This provides us with the full-flexibility of designing and implementing the blockchain with reliability and security as the first-class priorities.

- *Certified Consensus Protocol.* The blockchain that adopts CertiK Software utilizes Delegated Proof of Stake (DPOS) as the decentralized consensus algorithm. Block production is open to anyone who is certified to run a full note, convincing the community by receiving enough votes of CKT. Since the reputation of selected validators may drop, back-up validators may have the opportunity to replace former nodes. The implementation of the DPOS protocol will be verified using the CertiK verification platform. The blockchain that adopts the CertiK Software will award new gas, CKG, to a block producer every time a block is produced. This dual-coin model proposed by CertiK will be explained later in this section.
- *Certified Virtual Machine.* The blockchain that adopts CertiK Software will be equipped with a Certified Virtual Machine (CVM). CVM is based on, and backward compatible with, the Ethereum Virtual Machine (EVM). CVM treats reliability and security as its first-class concerns, and has two novel features: 1) the implementation of CVM will be fully formal verified and 2) unverified smart contracts will consume much more gas to run than verified ones. The hash of the proof (showing that the contract bytecode satisfies the specification) will be carried with the code, which prevents the consumption of CKG.

CertiK Verification Platform

The CertiK Software provides a verification toolchain to develop mathematical proofs of smart contracts and blockchain implementations. This toolchain is driven by the flow of

CKG.

- *Smart labeling.* The CertiK verification platform has designed a novel approach to create specifications of DApps/systems using labels. These labels are expressive enough to formally state the desired properties and are compatible with the existing programming languages (e.g., Solidity). By utilizing deep learning techniques with an expertly curated training set of labels, the CertiK verification platform intends to introduce a framework, named smart labeling, to understand decentralized programs – not only at the syntax level, but also at the semantics level — and automatically add proper labels to the source code.

- *Layer-based decomposition.* The CertiK team is among the first to achieve modular verification by realizing a novel concept, named layered deep specifications [1, 2, 3, 4]. This technique uncovers the insights of layered design patterns and makes it possible to decompose a complex proof task into smaller ones, while verifying each of them at their proper abstraction level.

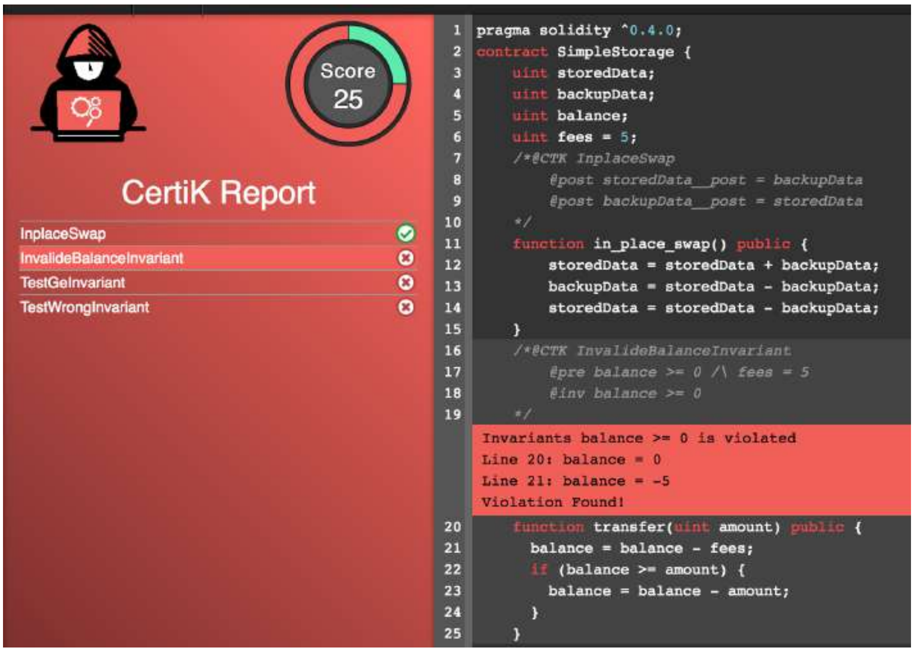
- *Pluggable proof engine.* These decomposed proof obligations are much easier to untangle and can even be solved by some automatic verifiers (e.g., SMT solvers [14]). To enable extensibility, the CertiK verification platform is intended to provide an open protocol, such that more advanced solving algorithms can be freely plugged into this system.

- *Machine-checkable proof objects.* The CertiK verification platform constructs mechanized proof objects (or counterexamples), such that these proofs can be quickly checked by anyone using their own machine. These proof objects can be viewed as the "certificates" [13] to the verified programs. The CeritK Gas CKG may be used to develop and validate the proof objects.

- *Certified DApp libraries.* In order to improve the code quality and reliability of the entire blockchain community, the CertiK verification platform offers a series of certified libraries and plug-ins to the integrated development environment (IDE) for building more trustworthy DApps. The use of these tools will cost a small amount of CKG as virtual crypto fuel/gas, but will provide more assurances during the development phase.

- *Customized certification services.* For DApps/systems (e.g., digital wallets) with high-reliability requirements, the CertiK verification platform intends to provide customized certification services. In this case, verification experts will help specify/verify the programs and generate a detailed, comprehensive report.

Figure 1 presents how to use the CertiK verification platform to verify two simple functions: in-place swap (line 7 to 15) and account transfer (line 16 to 25). Specifications to these functions can be expressed using CertiK labels, e.g., "@pre," "@post," and "@inv,"



- InplaceSwap ✓
- InvalideBalanceInvariant ☒
- TestGeInvariant ☒
- TestWrongInvariant ☒

```
1  pragma solidity ^0.4.0;
2  contract SimpleStorage {
3      uint storedData;
4      uint backupData;
5      uint balance;
6      uint fees = 5;
7      /*@CTK InplaceSwap
8          @post storedData__post = backupData
9          @post backupData__post = storedData
10     */
11     function in_place_swap() public {
12         storedData = storedData + backupData;
13         backupData = storedData - backupData;
14         storedData = storedData - backupData;
15     }
16     /*@CTK InvalideBalanceInvariant
17         @pre balance >= 0 /\ fees = 5
18         @inv balance >= 0
19     */
    Invariants balance >= 0 is violated
    Line 20: balance = 0
    Line 21: balance = -5
    Violation Found!
20     function transfer(uint amount) public {
21       balance = balance - fees;
22       if (balance >= amount) {
23         balance = balance - amount;
24       }
25     }
```

Figure 1: Screenshot of verifying "in-place swap" and "account transfer" methods using CertiK.

which represent pre-condition, post-condition, and invariants, respectively. Since these labels are written in the comments, there is no need to modify the compiler and no switching costs for developers. The functions, together with specifications (or labels), will then be processed and decomposed by the CertiK verification platform and sent to the verifier to solve. After the verification is compelte, the CertiK verification platform will return a detailed and comprehensive evaluation report. Counterexamples will be provided if the proof obligation cannot be satisfied. This allows for the immediate and transparent ability to fix identified vulnerabilities and re-run verification until the proof obligations are fully satisfied. As shown Fig. 1, the invariant that the balance is always non-negative is violated by the transfer function at line 21. With the CertiK verification platform's certified kits, feedback can be real-time, helping developers fix the discovered bugs during initial development.

Dual-Coin Model

PLEASE NOTE: CRYPTOGRAPHIC COINS REFERRED TO IN THIS WHITE PAPER REFER TO CRYPTOGRAPHIC COINS ON A LAUNCHED BLOCKCHAIN THAT ADOPTS THE CERTIK SOFTWARE. THEY DO NOT REFER TO THE ERC-20 COMPATIBLE TOKENS OR ANY OTHER TOKENS BEING DISTRIBUTED ON SOME BLOCKCHAINS IN CONNECTION WITH THE CKT/CKG DISTRIBUTION.

The blockchain that adopts the CertiK Software will have a dual-coin model. The native digital coin of the blockchain (CKT) is the key to decide which producer can be selected to

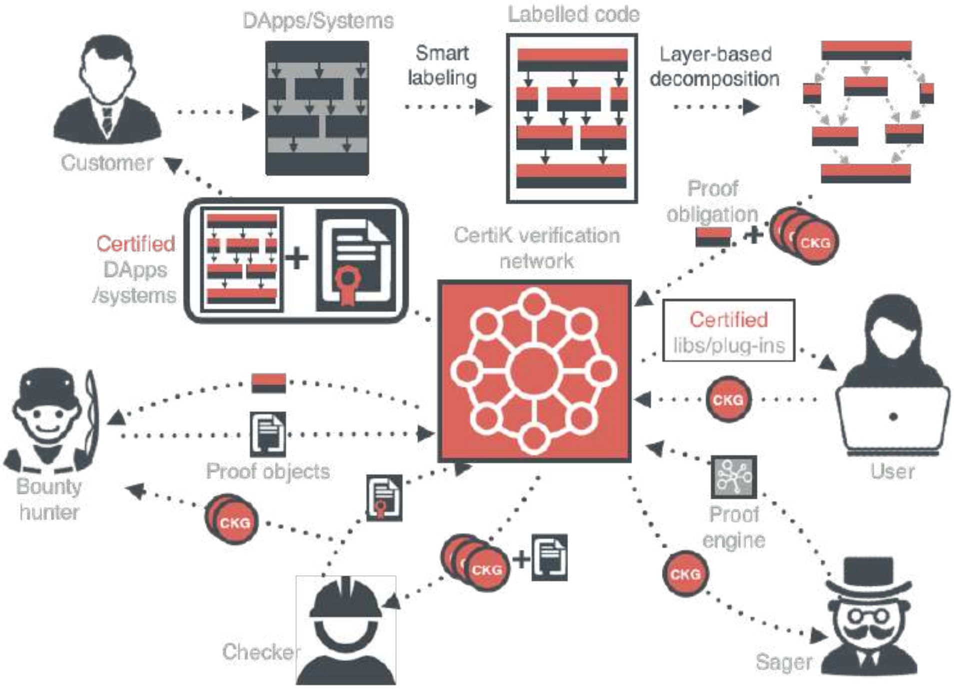


Figure 2: The CertiK framework and community.

generate blocks, and govern the entire network. CKT is the utility vote to reach consensus and solve disputes.

CKG is the utility gas and will be rewarded to the block producers. That is the only method in which CKG can be generated. This dual-coin model provides the following benefits: 1) preventing the additional offering of CKT; 2) separating transaction costs CKG from CKT; and 3) enabling an extensible usage of CKG.

CKT and CKG serve as non-refundable, functional utility fuel that will be used as both the unit of exchange between participants on the blockchain that adopts CertiK Software, as well as the economic incentives that will be consumed to encourage participants to contribute to, and maintain, the ecosystem (as described below). Both CKT and CKG are integral and indispensable parts of the CertiK Software, because in the absence of them, there would not be a common unit of exchange to reward and incentivise users for work done on the blockchain, thus rendering the ecosystem unsustainable.

The certified kits of the CertiK verification platform will be further powered by the CertiK community, creating a decentralized style of work, such that anyone can construct proofs, validate proofs, improve solver algorithms, and also contribute new proof obligations. The CertiK verification platform unifies the entire community by incorporating the flow of CKG within (but not limited to) five different roles (shown in Fig. 2):

- *Customers* may submit programs/systems that need verification (through the CertiK verification platform's services) or any proof obligations (that meet the open protocol) to the network. This is done by initiating and broadcasting a special "proof request" transaction associated with some CKG incentives offered for anyone who constructs the proofs.

- *Bounty hunters* aim for CKG incentives and offer to share their computation resources. They will construct and broadcast the proof objects, and then wait for the proofs to be validated. Due to the significance of this role, only users who possess a certain amount of CKG are allowed to take this role.
- *Checkers* may obtain CKG incentives by recording regular transactions or by checking the submitted proof objects. Bounty hunters may only receive their incentives once their proofs are validated, and checkers will also receive a small portion of these incentives.
- *Sages* plug in their proof engines via the CertiK verification platform's open protocol. Their engines may be randomly used by bounty hunters and will be evaluated through A/B testing. They can also receive some CKG incentives depending on the evaluation result of their engines. Outstanding engines will be studied and spread by the community.
- *Users* may subscribe to all CertiK verification platform's certified libraries and IDE plug-ins to build their own DApps/systems by using CKG.

These five roles will balance, guard, and improve the CertiK verification platform's community. Along with the CKG economy, real value is generated by posing and solving proof obligations, validating proof objects, and creating advanced proof engines. The Foundation believes that the proof/verification requirements are universal, which will keep the CertiK Software's community active.

Besides the verification platform, more use cases of CKG can be designed and implemented, e.g., using as the gas to access verified payment channels, etc.

Statement of CKT/CKG. CKT/CKG does not in any way represent any shareholding, participation, right, title, or interest in the Foundation, its affiliates, or any other company, enterprise or undertaking, nor will CKT/CKG entitle CKT/CKG holders to any promise of fees, revenue, profits, or investment returns, and are not intended to constitute securities in Singapore or any relevant jurisdiction. CKT/CKG may only be utilized on the CertiK/CKG Software, and ownership of CKT/CKG carries no rights, express or implied, other than the right to use CKT/CKG as a means to enable usage of and interaction within the CertiK Software. In particular, you understand and accept that CKT/CKG:

(a) is non-refundable and cannot be exchanged for cash (or its equivalent value in any other virtual currency) or any payment obligation by the Foundation or any affiliate;

(b) does not represent or confer on the CKT/CKG holder any right of any form with respect to the Foundation (or any of its affiliates) or its revenues or assets, including without limitation any right to receive future revenue, shares, ownership right or stake, share or security, any voting, distribution, redemption, liquidation, proprietary (including all forms of intellectual property), or other financial or legal rights or equivalent rights, or intellectual property rights or any other form of participation in or relating to the CertiK Software, the Foundation, the Distributor and/or their service providers;

(c) is not intended to be a representation of money (including electronic money), token, security, commodity, bond, debt instrument or any other kind of financial instrument or investment;

(d) is not a loan to the Foundation or any of its affiliates, is not intended to represent a debt owed by the Foundation or any of its affiliates, and there is no expectation of profit; and

(e) does not provide the CKT/CKG holder with any ownership or other interest in the Foundation or any of its affiliates.

The contributions in the CKT distribution will be held by the Distributor (or its affiliate) after the CKT distribution, and contributors will have no economic or legal right over or beneficial interest in these contributions or the assets of that entity after the CKT distribution.

To the extent a secondary market or exchange for trading CKT/CKG does develop, it would be run and operated wholly independently of the Foundation, the Distributor, the sale of CKT and the CertiK Software. Neither the Foundation nor the Distributor will create such secondary markets nor will either entity act as an exchange for CKT/CKG.

2 Market Analysis

In the past two years, due to the considerable popularity of Bitcoin, blockchain technology and DApps have become increasingly popular worldwide. The soaring price of cryptocurrencies is likely derived from the exponential growth in the number of smart contracts and other DApps. As of January 2018, there were more than one million contracts deployed on Ethereum [20], a stark increase from the 0.12 million contracts deployed on Ethereum a year and a half prior [21]. Many people [22] believe that blockchain will fundamentally change several aspects of society. The development infrastructure has continued to improve, and this significant growth of deployed smart contracts is expected to continue. Based on the current growth rate, the total amount of such DApps may reach 10 million in the next few years.

Market size estimation. The global blockchain market is forecasted to grow from $1.2 billion in 2018 to $23.3 billion in 2023 at a compound annual growth rate (CAGR) of 80.2% [28]. This growth has been catalyzed by the multitude of smart contract use cases across various industry verticals. Because smart contracts are self-executing, open-sourced, and permanent, their reliability and security are especially vital. As a result, there has been a high demand for verification services across all use cases, especially those that involve virtual currencies. Today, the majority of smart contracts have been developed for the purpose of minting tokens or conducting fairly simple logic, and existing smart contract verification service providers may charge tens of thousands to hundreds of thousands of dollars for a single audit, although their techniques cannot adequately address the challenges mentioned in Sec. 1.. Over time, however, it is likely that developers may begin to create more complex smart contracts that handle sophisticated responsibilities which can truly disrupt the status quo. For these smart contracts, formal verification and absolute confidence will be even more important when conducting an audit. It is evident that this kind of verification service would be highly profitable, with a high barrier to entry. At the rate in which blockchain is expected to expand in the next few years, an average contract value (ACV) of one hundred thousand dollars would illustrate a market that can be as large as one trillion dollars (i.e., $10M \times \$100K$).

Moreover, the techniques used in the CertiK Software are applicable to a broader market than solely providing the auditing services that existing providers offer. With certified DApp libraries and the IDE plug-ins, the CertiK Software is able to provide real-time and interactive verification feedback. These services may reduce the development costs by shortening the development cycle and replacing some of the testing infrastructure. According to collected data, the testing development and maintenance take about 40% of the total development cost [23] and, a DApp with a high quality service typically takes around one million dollars to build. In this sense, if the CertiK Platform certified libraries and plug-ins replace 20% of the testing infrastructure, it will be a trillion-dollar market (i.e., $20\% \times 40\% \times 10M \times \$1M$).

Potential competitors. CertiK stands at a unique position in the market by offering rigorous formal verification, which was proven capable of verifying extremely complex systems with a labeling approach, while also achieving scalability with a layer-based decomposition approach. These advantages of thoroughness and speed have been heavily desired in the blockchain developer community, but with existing manual approaches to auditing smart contracts, the two qualities had been seen as paradoxical. Quantstamp [24] proposes a verification protocol for smart contracts written in Solidity. It utilizes the traditional model checking techniques, as opposed to full formal verification, and requires an intensive amount of human effort for reviewing the source code and writing the specification manually. This limits the scalability of their approach. Also, it is unclear how to extend the Quantstamp techniques to verify complex systems like the blockchain itself. Despite these limitations, the estimated valuation of Quantstamp is more than $350 million. Solidified [25] and Securify are two other companies claiming to provide verification services for smart contracts. Their services can only be used to check/verify a list of fixed properties rather than the functional correctness. Again, it is unclear how these techniques can verify complex systems, given the limitations in scalability.

Zeppelin is a testing/verification service provider that is most widely known from their open-source framework for smart contract development, named OpenZeppelin. The libraries provided by OpenZeppelin are used by many as the foundation of basic smart contracts. However, these libraries are most useful for simple smart contracts because complex contracts are heavily customized. Additionally, these libraries are not fully verified and/or do not offer mechanized proof objects.

Runtime Verification is a traditional formal verification company that had begun providing verification services for smart contracts. Similar to the CertiK team, they also have a strong academic background and have proposed a semantic of the EVM, named KEVM. The cornerstone of their methods requires developers to code within a specific framework, namely the K Framework, in order to undergo formal verification. This increased friction stifles mass adoption, as it is dependent on developers changing their coding practices. Additionally, much of their work still remains at the research stage. It is still a big unknown as to whether their techniques can be applied to industry-grade, complex systems.

3 CertiK Technical Toolchain

This section uses the notorious TheDAO attack as an example to explain the sequence of techniques used in the CertiK verification platform. Figure 3 shows the pseudocode to replay TheDAO attacks. The vulnerable bank contract (lines 1 to 19) maintains the account balance; it adds the deposit value to the balance and reduces the balance with the withdrawal value. However, there is a server bug, in which the server first sends the assets (at line 14) before updating the balance (at line 17) in the withdrawal method. Thus, attackers (lines 20 to 32) can utilize the fallback and synchronization features of smart contracts to perform "multiple spend attacks". The attacker begins by calling the withdraw method (at line 29). When the assets are sent back (at line 14), the fallback function (at line 23) of the attacker is triggered, and another withdraw method is invoked again. Due to the fact that balance has not been updated yet, the bank will issue another send. This simple issue of TheDAO digital wallet has allowed the theft of more than $50 million worth of Ether.

CertiK labeling. In order to detect and prevent these types of bugs, one must be able to precisely specify the expected behaviors of this bank contract. This can be achieved by using CertiK labels, which are lightweight, yet expressive.

Consider the bank contract example. Its specification can be merely written as a single equation: "balance = deposit - withdraw." Thus, one can insert "@pre" (at line 7), and "@post" (at line 8) labels with this equation before all the method declarations to ensure

```
/* The vulnerable bank contract */
contract Bank{
  uint balance;
  function depositBalance() {
    balance = balance + msg.value;
  }
  /*@CERTIK TestBalance
      @pre balance = deposit - withdraw
      @post balance_post = deposit_post - withdraw_post
      @inv balance <= deposit - withdraw
      @fun "": lambda v. withdraw_post = withdraw + v
  */
  function withdrawBalance() {
    if (msg.sender.call.value(balance)() == false) {
        throw;
    }
    balance = 0; /* BUG! Reduce balance after sending money */
  }
}
/* The malicious contract that attacks the bank*/
contract Attacker {
  address bankAddress;
  function() { /* fallback function call withdrawnBalance recursively */
    bankAddress.call(bytes4(sha3("withdrawBalance()")));
  }
  function deposit() {
    bankAddress.call.value(2).gas(20764)(bytes4(sha3("depositBalance()")));
  }
  function withdraw() { /* triggers withdrawBalance in the contract Bank*/
    bankAddress.call(bytes4(sha3("withdrawBalance()")));
  }
}
```

Figure 3: Pseudocode illustrating TheDAO attack.

that this specification is satisfied. Figure 3 shows the labels for the withdrawal method and uses the postfix "_post" to represent the value after the method's execution.

However, merely ensuring this equation before and after the method call is not strong enough. Temporal breakdown of this equation during the method call may lead to severe consequences. The control flow can be exploited by malicious fallback functions, and this weak point may make the entire contract vulnerable. To solve this issue, one can insert a "@inv" label, which means that the followed property holds at any point of the execution. One weakens the equation to the invariant "balance $\leq$ deposit - withdraw", such that these values do not have to be updated at the same time. Since the fallback function is a "black box" to the bank contract, one can only utilize the "@fun" label to specify its known effects: the total assets are equal to the balance that has been withdrawn.

Using this label-based language, the specification of this bank contract can be easily expressed in a formal and comprehensive way. CertiK labels can be used to write any properties in the first class logic. The Foundation plans to add higher-order support shortly, which will allow these labels to be rich enough to specify almost all deployed smart contracts, DApps, and blockchain systems.

This label-based language is designed such that it is entirely possible to label source programs automatically. The Foundation intends to establish a large training set containing certified DApp libraries expertly curated by the CertiK team from selected DApps of popular domains. The Foundation intends to apply deep learning techniques to build the smart labeling framework. With this framework, most of the shared logic and properties can be automatically labeled, dramatically reducing the specification and proof efforts can be dramatically required.

CertiK proof engine. The labeled programs will then be compiled using CertiK compiler, developed in-house by formal verification experts. Different from the general-purpose compiler, CertiK compiler recognizes the label language and can parse the labeled programs into an internal model for DApps. This model can be viewed as abstract automata defining how the DApps will change the system state (consisting of all global and local variables). This model is language-independent, such that the back-end of the proof engine can be unified.

The proof obligation, which states whether the program behavior executing on the CertiK verification platform's internal DApp model meets the program's specification (generated from labels), can be converted into a set of constraints. Take the invariant proof for the program in Fig. 3 as an example. At the beginning of the withdraw function (ar line 12), the constraint for the @pre label is generated, i.e., "C1: balance = deposit - withdraw". One must validate the invariant defined using the @inv label by checking if "I2: $\neg$(balance $\leq$ deposit - withdraw)" can be satisfied. Since I2 is always false given C1, one knows the invariant holds at the beginning. Then, at line 13, the function call of the message sender increases the withdraw value with the current balance. For such value updates, one can introduce a new version of the variable, e.g., "withdraw_1" and encode the update into the constraints “C2: withdraw_1 = withdraw + balance”. To validate the invariant at this point, one may check if “I2: ¬(balance ≤ deposit - withdraw_1)” can be satisfied. Here, given C1 and C2, one may have that “deposit - withdraw_1 = 0”. Thus, any balance with positive initial value will break the invariant. In this way, the withdrawal bug that caused the TheDAO attack could have beeneasily detected using the CertiK proof engine.

This procedure can be done by SMT solvers [14], and counterexamples (or hints) will be generated if the problem can be solved. The soundness of the counterexamples (or the proof) can be easily checked with respect to the proof obligation, which forms the basis of the Proof-of-Proof mechanism.

Furthermore, to improve the performance of this solving procedure, the Foundation intends to design an open protocol such that any SMT solvers can be plugged into the CertiK Platform’s network. These solvers will be randomly selected to prove some already verified programs and the results, including the execution time and the quality of the generated hints, will be evaluated. The solvers with better performance have a higher chance to be selected.

Layer-based decomposition As explained previously, SMT solvers often encounter the state explosion problem [12] when dealing with complex systems. This means that, as the complexity of the system increases, the state space to explore becomes too large to cover. To address this issue and apply CertiK verification platform’s techniques to a broader domain, the Foundation intends to introduce a novel layered-based approach. By developing the programming language support for building and composing layer-based specifications, the CertiK verification platform enables a disciplined way of decomposing a complex system into a large number of small components and proof obligations. Without using layers, one might have to consider arbitrary interactions between the current component and its environment: an invariant held in one function can be easily broken when it calls a function defined in another module or communicates with another entity. A layered approach aims to sort and isolate all components based on a carefully designed set of abstraction levels so one can reason about one small abstraction step at a time. This can dramatically simplify the environment model that needs to be considered at each layer. In the past, the founders of the CertiK team had successfully built the world’s first fully verified, concurrent OS kernel, named CertiKOS [1, 2], using the Coq proof assistant [26]. CertiKOS consists of 6,500 lines C and assembly and is divided into more than 60 layers. The whole proof efforts are only about two person-years.

Based on these successes, the CertiK team plans to create a new layered-based verification framework that is suitable to be used to reason about blockchain ecosystems. The key idea is to model the behaviors of the environment, including the context contracts and context nodes, in a compositional way. Recall TheDAO attack example, the root of that bug is the neglect of the context’s fallback functions. By proving that each layer component meets its layer specification under arbitrary context and linking all the proofs together, the end-to-end correctness of the entire system can be guaranteed.

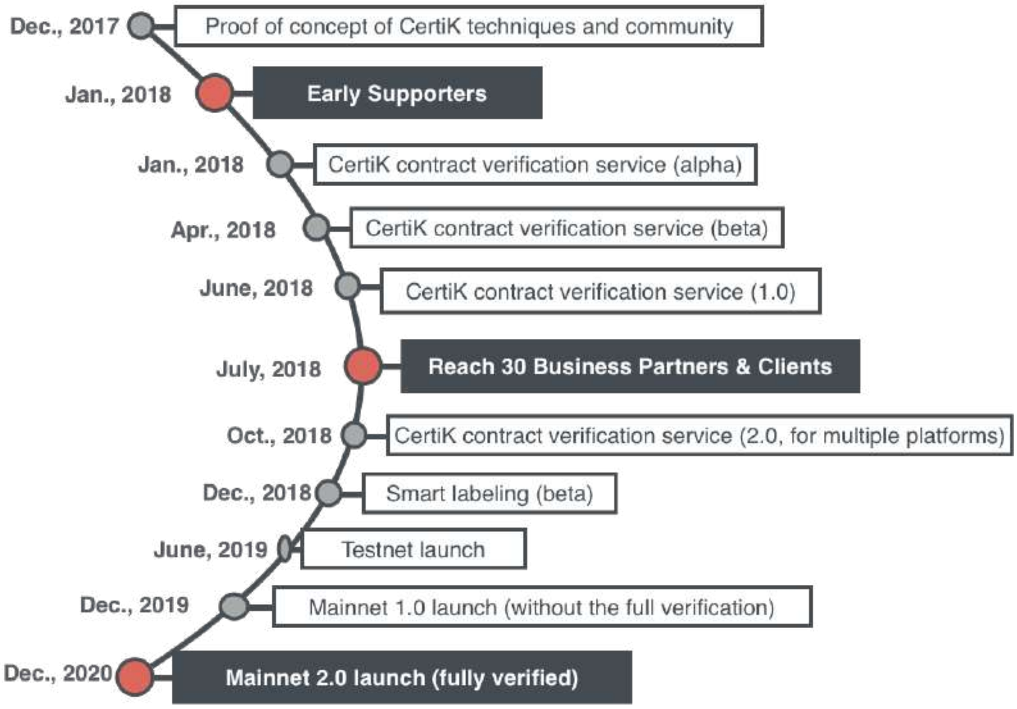

Dec., 2017
Proof of concept of CertiK techniques and community
Jan., 2018
Early Supporters
Jan., 2018
CertiK contract verification service (alpha)
Apr., 2018
CertiK contract verification service (beta)
June, 2018
CertiK contract verification service (1.0)
July, 2018
Reach 30 Business Partners & Clients
Oct., 2018
CertiK contract verification service (2.0, for multiple platforms)
Dec., 2018
Smart labeling (beta)
June, 2019
Testnet launch
Dec., 2019
Mainnet 1.0 launch (without the full verification)
Dec., 2020
Mainnet 2.0 launch (fully verified)

Figure 4: The roadmap of the CertiK Software.

4 Roadmap, CKT/CKG Distribution, and Budget Plan

4.1 Roadmap

Figure 4 shows the roadmap of the CertiK Software. The proof of concept of the CertiK Software's techniques and community began in December 2017. The product development plan was quite aggressive. The Foundation planed to launch the alpha version of CertiK smart labeling and the layered verification techniques by the end of 2018. These prototypes were illustrated by online demos provided by the CertiK team. To demonstrate the power of its approach, the Foundation aimed to establish business partnerships and clients with at least ten organizations (or projects) in the blockchain community once the beta version of the CertiK verification platform was launched in April 2018. These strategic supporters would be further expanded to reach at least 30 partners and clients by the end of July 2018. The Foundation then focused on developing new verification techniques, maintaining the CertiK Software's community, and spreading this idea of decentralized certification.

By the end of February 2019, the CertiK Software fulfilled all the milestones set in the previous roadmap. Figure 5 shows select partners and clients of the CertiK Software. The CertiK team partnered as the required/recommended auditor of the largest cryptocurrency exchanges, including Binance, OKEx, Huobi, Bittrex, Kucoin, and more. The CertiK verification platform has been used to verify more than 100 smart contracts, including

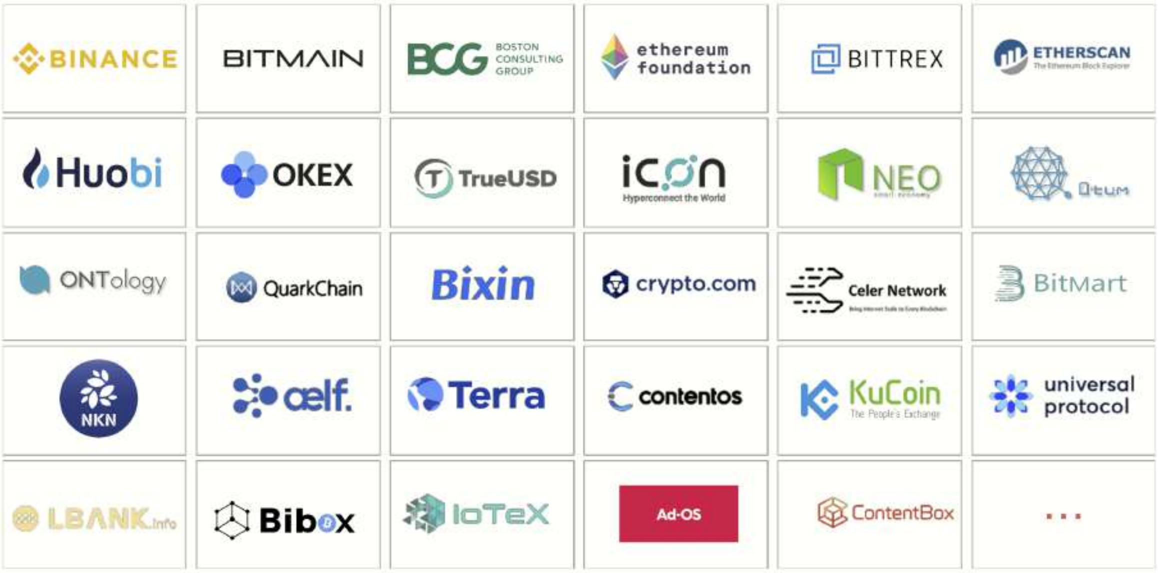


Figure 5: Selected partners and clients the CertiK Software.

those of BNB, TrueUSD, Crypto.com, Quarkchain, Celer Network, etc. Over $1.2 billion worth of cryptocurrency assets have been now secured by the CertiK Software.

The CertiK team plans to launch the testnet that adopts the CertiK Software by the end of June 2019. The CertiK team plans to launch the mainnet 1.0 (without the full verification) that adopts the CertiK Software by the end of 2019, and the mainet 2.0 (with the full verification) by the end of 2020. Timelines are subject to change, but the team intends to continue to pursue and exceed the aggressive roadmap that it has been setting forth.

4.2 Dual-Coin Metrics

This section outlines the numbers regarding CKT/CKG distribution. The Foundation believes that transparency of CKT/CKG metrics will be helpful for maintaining a healthy community. The distributor of CKT/CKG (the Distributor) shall be an affiliate of CertiK Foundation.

CKT Distribution

The total supply of CKT will be capped to 100 million (100,000,000) and distribution metrics are shown in Figure 6(a), where 35% will be distributed to accredited investors and 5% of CKT will be directly offered through a digital asset exchange.

The contributions in the CKT distribution will be held by the Distributor (or its affiliate), and contributors will have no economic or legal right over or beneficial interest in

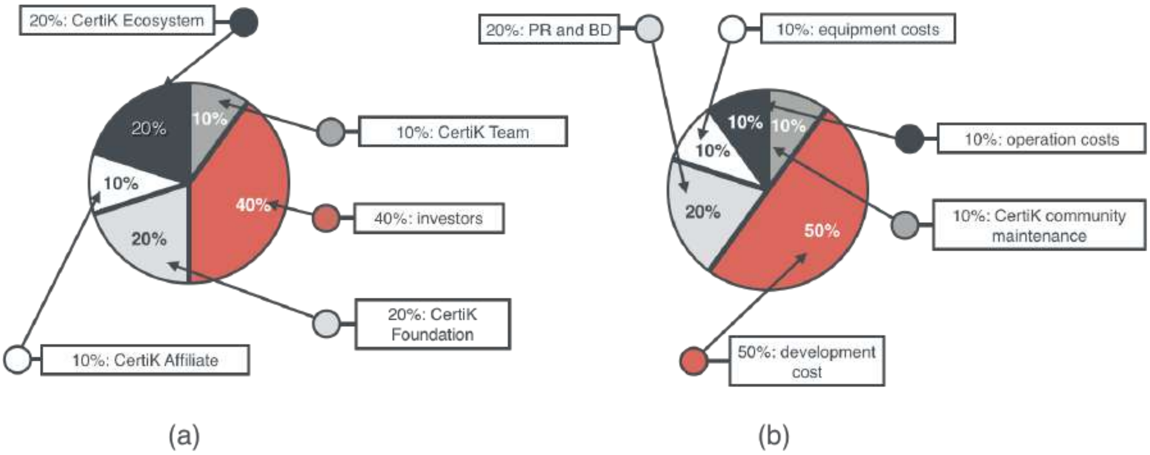


Figure 6: (a) CKT distribution metrics and (b) budget plan.

these contributions or the assets of that entity after the CKT distribution. To the extent a secondary market or exchange for trading CKT does develop, it would be run and operated wholly independently of CertiK Foundation, the Distributor. Neither the Foundation nor the Distributor will create such secondary markets nor will either entity act as an exchange for CKT.

CertiK Foundation will reserve 20% of CKT to vote for super nodes in the network, which cannot be transferred. This portion will be unlocked to vote linearly over a 3-year period. This reservation will play a positive role in maintaining a healthy network.

CertiK Foundation and its affiliates will reserve another 10% of CKT as rewards for the development and maintenance of the CertiK verification platform. CertiK team will reserve 10% of CKT. The rest of the 20% of CKT will be used as rewards and payments for contributors/collaborators of the CertiK ecosystem. All the above allocations will be vested over a 3-year period. Upon the expiry of one year, they will receive one third (i.e., 1/3) of this allocation, with an additional 1/3 of these allocations to be distributed upon the expiry of each year thereafter, until the 3-year period is up.

CKG Generation

CKG can only be generated as rewards to produce blocks. The total supply of CKG will be capped at 1 billion (1,000,000,000). The ratio of CKT and CKG will be 1: 10. All the CKG will be gradually mined in 16 years.

The CKG mined by CertiK Foundation directly will be used as bounty programs, community incentives and marketing initiatives in order to promote a larger user base. This portion of CKG will also cover expenditures in areas of legal, KYC/AML, consulting services, operational costs, IT software/hardware, ecosystem support, expansion of CertiK network infrastructure. The board of CertiK Foundation will manage the use of resources related to CKG, for the best of CKG users, participants, and stakeholders.

4.3 Budget Plan

Figure 6(b) shows how the capital will be utilized to develop CertiK techniques and grow the community. At the early stage, the software development will take up the most significant portion of the budget. Research, legal, and financial consulting are also necessary. Thereafter, once the alpha version is complete, the Foundation will allocate more resources to building and maintaining the CertiK community. The Foundation will continuously provide tutorials about the CertiK Software's services and provide detailed instructions for developers/miners/users about how to participate in the verification process. The Foundation plans to make a series of technical talk videos about the CertiK Software and maintain them on social media. Further, the Foundation plans to utilize its academic resources to give lectures, hold seminars, and even organize summer schools about how to build trustworthy blockchain ecosystems.

In addition, the Foundation plans to build a strong development team by hiring at least 20 software engineers and research scientists. The Foundation will keep evolving the CertiK Software, making sure its technologies are always leading the market.

5 Team Leaders

Prof. Zhong Shao (Co-Founder)
Chair of Computer Science Department, Yale University
Thomas L. Kempner Professor, Yale University

Zhong Shao is Thomas L. Kempner Professor and Department Chair of the Department of Computer Science at Yale University. He earned his Ph.D. in Computer Science from Princeton University in 1994. During his early career, he was a key developer of the SML/NJ compiler and the main architect of its FLINT certifying infrastructure. In recent years, Prof. Shao has been a leading figure working on the highly visible research fields on cybersecurity, programming languages, operating systems, and certified software. He and his FLINT group at Yale have developed the world's first hacker-resistant, concurrent operating system CertiKOS—a major milestone toward building cyber-physical systems that are provably free from software vulnerabilities. Shao is the author or co-author of 90 articles in top scientific journals and conferences.

Prof. Ronghui Gu (Co-Founder)
Assistant Professor, Columbia University

Ronghui Gu is a tenure-track Assistant Professor of Computer Science at Columbia University. He obtained his Ph.D. in Computer Science from Yale University in 2016, where his dissertation won the Distinction Dissertation Award at Yale and was nominated for the ACM Dissertation Award. He obtained his B.S. from Tsinghua University in 2011. Prof. Gu is an expert in formal verification of system software. He was the primary designer and developer of CertiKOS, the world's first fully verified concurrent OS kernel. His OSDI16 paper on CertiKOS has been nominated and selected for publication in the Research Highlights section of the CACM.

Dr. Vilhelm Sjöberg (Principal Scientist)
Ph.D. in Computer Science, University of Pennsylvania
John C. Reynolds Doctoral Dissertation Award Winner, 2016

Vilhelm Sjöberg is a former associate research scientist at Yale University. He received his Ph.D. in Computer Science from the University of Pennsylvania in 2015. He is an expert in software verification, programming languages, and type systems. Currently he is interested in language support for layered verified systems like CertiKOS. Dr. Sjöberg is the winner of 2016 ACM SIGPLAN John C. Reynolds Doctoral Dissertation Award.

Dr. Zhongzhong Ni (VP of Engineering)
Ph.D. in Computer Science, Yale University
Ex-Engineering Lead at Google, and researcher at Microsoft Research

Zhaozhong Ni was formerly an engineering lead at Google and HP / 3PAR and a researcher at Microsoft Research. He is an expert in systems software and formal verification and has extensive experience in building operating systems kernels and mission-critical enterprise systems. Dr. Ni was a founding member of gVisor, Google's new security-focused OS with cloud-scale production. He holds multiple patents on distributed storage systems. He obtained his BS from Tsinghua University in 2000 and his PhD in Computer Science from Yale University in 2006.

Daryl Hok (COO)
Dual Bachelors degree in Economics and Psychology, Yale University

Daryl Hok spearheaded Corporate Development at FiscalNote, a global machine-learning legal tech company, where he accelerated growth by completing 3 acquisitions in 12 months, including a $180M purchase from The Economist Group. He was also the Product Manager responsible for ideation and release of a SaaS product with several million USD in recurring revenue, along with the development of the core data infrastructure. He obtained a dual BA in Economics and Psychology from Yale University, with concentrations centered around behavioral economics.

Yvan Nasr (Head of Business Development)
MBA, Booth School of Business, University of Chicago

Yvan Nasr led various Product and Business Development initiatives at Samsung Electronics and was the driving force behind Barclays Banks digital transformation in key European markets - UK, Spain, Italy. Prior to obtaining his MBA, Yvan also led the growth of Europes largest retail holding, Kingfisher PLC, diversifying the brands global footprint and product ranges in key Middle Eastern and Asian markets. Recently, he was the Head of Partnerships at Hosho, a blockchain cybersecurity startup. He obtained his MBA from the University of Chicago, Booth School of Business.

Muhan Zou (EVP and CSO)
M.S. in Computer Science, Yale University
Ex-Engineering Lead at Comcast, Oracle

Muhan Zou serves as the founding member of CertiK from company inception, where he contributes with both his engineering abilities and team leadership. Muhan has years of experience in designing and developing enterprise level SaaS products. Prior to joining, he worked as the engineering lead at Comcast to monetize large-scale ads and set-top-box/linear raw data into business insights deliverables. He also worked as a full-stack engineer at Oracle where he built the social cloud platform after obtaining his MA in Computer Science from Yale University.

Dr. Kai Yan (CBO)
Ph.D. in Economics, Yale University
Advisors: Gary Gorton, Robert Shiller (Nobel Prize in Ecomonics, 2013)

Dr. Kai Yan has extensive experience in business growth, economic research and financial markets. He previously worked as an economist at the International Monetary Fund and a strategist at a hedge fund. At the IMF, he worked with global regulators to devise new regulations around an improved banking system. While working at a hedge fund, he was responsible for designing and executing trading strategies. Dr. Yan obtained his BA in Economics from Peking University and his PhD in Economics at Yale University.

6 Risks

You acknowledge and agree that there are numerous risks associated with purchasing CKT/CKG, holding CKT/CKG, and using CKT/CKG for participation in the CertiK Software.

6.1 Uncertain Regulations and Enforcement Actions

The regulatory status of CKT/CKG and distributed ledger technology is unclear or unsettled in many jurisdictions. It is impossible to predict how, when, or whether regulatory agencies may apply existing regulations or create new regulations with respect to such technology and its applications, including CKT/CKG and/or the CertiK Software. Regulatory actions could negatively impact CKT/CKG and/or the CertiK Software in various ways. The Foundation (or its affiliates) may cease operations in a jurisdiction in the event that regulatory actions, or changes to law or regulation, make it illegal to operate in such jurisdiction, or commercially undesirable to obtain the necessary regulatory approval(s) to operate in such jurisdiction.

After consulting with a wide range of legal advisors and continuous analysis of the development and legal structure of virtual currencies, the Foundation will apply a cautious approach towards the distribution of CKT/CKG. For the sale of CKT/CKG, the Foundation is working with Tzedek Law LLC, a boutique corporate law firm in Singapore with a good reputation in the blockchain space.

6.2 Loss of Talent

The development of the CertiK Software depends on the continued co-operation of the existing technical team and expert consultants, who are highly knowledgeable and experienced in their respective sectors. The loss of any member may adversely affect the CertiK Software or its future development.

6.3 Failure to Develop

There is the risk that the development of the CertiK Software will not be executed or implemented as planned, for a variety of reasons, including without limitation, the event of a decline in the prices of any digital asset, virtual currency or CKT/CKG, unforeseen technical difficulties, and shortage of development funds for activities.

6.4 Other Risks

In addition to the aforementioned risks, there are other risks (as more particularly set out in the Terms and Conditions) associated with your purchase, holding and use of CKT/CKG, including those that the Foundation cannot anticipate. Such risks may further materialise as unanticipated variations or combinations of the aforementioned risks. You should conduct full due diligence on the Foundation, its affiliates and the CertiK team, as well as understand the overall framework and vision for the CertiK Software prior to purchasing CKT/CKG.

References

[DeepSpec] DeepSpec: The science of deep specifications. http://deepspec.org/.

[YaleNews] CertiKOS: A breakthrough toward hacker-resistant operating systems. *Yale News*, 2016.

[IBTimes] CertiKOS: Yale develops world's first hacker-resistant operating system. *Internaltional Business Times*, 2016.

[YDN] Yale computer scientists unveil new OS. *Yale Daily News*, 2016. .

[1] R. Gu, J. Koenig, T. Ramananandro, Z. Shao, X. Wu, S. Weng, H. Zhang, and Y. Guo. "Deep specifications and certified abstraction layers." In *42nd ACM Symposium on Principles of Programming Languages (POPL'15)*.

[2] R. Gu, Z. Shao, H. Chen, X. Wu, J. Kim, V. Sjöberg, and D. Costanzo. "CertiKOS: An Extensible Architecture for Building Certified Concurrent OS Kernels." In *12th USENIX Symposium on Operating Systems Design and Implementation (OSDI'16)*.

[3] D. Costanzo, Z. Shao, and R. Gu. "End-to-end verification of information-flow security for C and assembly programs." In *37th ACM SIGPLAN Conference on Programming Language Design and Implementation (PLDI'16)*.

[4] H. Chen, X. Wu, Z. Shao, J. Lockerman, and R. Gu. "Toward compositional verification of interruptible OS kernels and device drivers." In *37th ACM SIGPLAN Conference on Programming Language Design and Implementation (PLDI'16)*.

[5] S. Nakamoto. "Bitcoin: A peer-to-peer electronic cash system." http://bitcoin.org/bitcoin.pdf.

[6] Wood, G., 2014. Ethereum: A secure decentralised generalised transaction ledger. Ethereum Project Yellow Paper, 151.

[7] G. Klein, K. Elphinstone, G. Heiser, J. Andronick, D. Cock, P. Derrin, D. Elkaduwe, K. Engelhardt, R. Kolanski, M. Norrish, T. Sewell, H. Tuch, and S. Winwood. "seL4: Formal verification of an OS kernel." In 22nd ACM Symposium on Operating Systems Principles (SOSP 09).

[8] E. W. Dijkstra. "Notes on structured programming." In Structured programming, pages 182. Academic Press, 1972.

[9] S. Peters, A. Danis, K. Elphinstone, and G. Heiser. "For a microkernel, a big lock is fine." In Asia Pacific Workshop on Systems (APSys 15).

[10] M. von Tessin. "The Clustered Multikernel: An Approach to Formal Verification of Multiprocessor Operating-System Kernels." PhD thesis, School of Computer Science and Engineering, The University of New South Wales, March 2013.

[11] Clarke, Edmund M., Orna Grumberg, and Doron Peled. "Model checking." MIT press, 1999.

[12] McMillan, Kenneth L. "Symbolic model checking." In Symbolic Model Checking, pp. 25-60. Springer, Boston, MA, 1993.

[13] Shao, Zhong. "Certified software." Communications of the ACM 53, no. 12 (2010): 56-66. Harvard

[14] De Moura, Leonardo, and Nikolaj Bjrner. "Z3: An efficient SMT solver." Tools and Algorithms for the Construction and Analysis of Systems (2008): 337-340.

[15] https://github.com/ethereum/go-ethereum/issues.

[16] King, Sunny, and Scott Nadal. “Ppcoin: Peer-to-peer crypto-currency with proof-of-stake.” self-published paper, August 19 (2012).

[17] Buterin, Vitalik. “A next-generation smart contract and decentralized application platform.” white paper (2014).

[18] http://heartbleed.com/.

[19] V. Buterin. Critical update re: Dao vulnerability, 2016.

[20] https://etherscan.io/accounts/c, Jan, 2018.

[21] How many contract are currently deployed on the ethereum blockchain?, September, 2016.

[22] IBM Sees Blockchain as the Next Big Thing.

[23] Application testing costs set to rise to 40% of IT budget.

[24] Quantstamp whitepaper.

[25] Solidified whitepaper.

[26] The Coq proof assistant. The Coq development team. http://coq.inria.fr.

[27] John C. Reynolds Doctoral Dissertation Award, 2016.

[28] Blockchain Market worth $23.3 billion by 2023, 2018.