# ZEN AUTHORIZED PROTOCOL

## PREAMBLE

This document definesthe Zen authorized protocol (the "**Authorized Protocol**"), including conditions that users must meet in order to use the Licensed Software pursuant to the Software License. For the avoidance of doubt, nothing other than the Software License grants you permission to use, propagate or modify the Licensed Software, and the Authorized Protocol sets additional conditions that must be met in order to use the Licensed Software in compliance with the Software License. You are not required to accept the Authorized Protocol; however, failure to accept and act in accordance with the Authorized Protocol terminates your right to use the Licensed Software. Use of the Licensed Software in contravention of either the Authorized Protocol or the Software License is considered a breach of the Software License and copyright infringement.

You may not modify the Authorized Protocol or the Licensed Software except as expressly provided under the Authorized Protocol. Any attempt otherwise to modify the Authorized Protocol or the Licensed Software is considered a breach of the Software License, automatically terminates your right to use the Licensed Software and is copyright infringement.

By using the Licensed Software, you indicate your acceptance of the Authorized Protocol and Software License.

All service providers or developers who produce tools to facilitate the construction and signing of smart contracts or transactions on behalf of other Zen Token holders must present this Authorized Protocol document and other referenced agreements. Service providers and developers shall be liable for losses resulting from failure to disclose the full terms to users.

Terms used herein but not otherwise defined have the meanings assigned to them in the Software License.

Up until now you have had access to software due to paying money as described in the Software License.
This document supplements the old one as part of the community release.

This software is in the copyright of Zen Protocol LTD (Owned by Blockchain Development LTD). While the company is not waiving its rights or IP in the software, it is granting some rights and opening it up to anyone who respects the following rules. In other words, the company waives its ability to censor, limit, revoke, etc any use of the system so long as you follow the rules.

## ARTICLE I

## AUTHORIZED PROTOCOL

Section 1.01 Authorized Protocol. The Authorized Protocol is the set of rules and processes by which blocks are added to the Community Release Blockchain in a manner that the Community Release does, always linked to the Genesis Block. The Authorized Protocol includes the version of the Licensed Software that accepts those blocks and transactions that the Community Release accepts and rejects those blocks and transactions which the Community Release rejects.

Section 1.02 Community Release. The Community Release is the later of (i) the first protocol client compiled from the code with git commit/tag identifier [37ce6b3f626137b09147849b146716fbf5ad4872] released on June 30, 2018 (also known as "**Malkuth**"), and (ii) any subsequent version of code that is approved by Community Vote. The Community Release shall always tie back to the Genesis Block.

Section 1.03 Genesis Block. The double SHA-3 Hash of the Genesis Block that initiates the Community Release Blockchain (the block with block number 1 accepted by the Community Release), dated June 30th, 2018, is attached in Appendix A below.

Section 1.04 Initial Circulating Supply; Increase in Supply. The Initial Circulating Supply of Zen Tokens[1] (i.e., those Zen Tokens that have been issued on the Community Release Blockchain) is 20% of the total number of Zen Tokens to be issued. Generally, additional Zen Tokens shall be generated according to the following approximate sexennial schedule:

a. Over the six years following the Community Release Date, 40% of the total number of Zen Tokens to be issued shall be issued;
b. Over the following six years, an additional 20% of the total number of Zen Tokens to be issued shall be issued;
c. Over the third sexennial period, an additional 10% of the total number of Zen Tokens to be issued shall be issued; and
d. For each successive sexennial period thereafter, half as great a percentage as was applied in the preceding period shall be applied.

## ARTICLE II

## SOCIAL CONSENSUS

[1] Zen Tokens mean the native Zen Blockchain tokens accepted by the Authorized Protocol and used for contract activation.

Section 2.01 Community. Miners neither govern nor determine the overall development of the project. Rather, the rules are made by the community of Zen Token holders. To this end, Article II provides rules for decentralized governance in using the Licensed Software. The following principles underpin this document:

a. Decisions shall not be determined on the basis of 'one CPU one vote' or any other hashpower mechanism.
b. Decisions shall not be made by possession or control of any single code repository.
c. Decisions shall not be made by hard forks or soft forks.
d. Decisions shall not be made by blackmail, rants on social media, intimidation or any other coercive means.
e. Rather, decisions shall be made by a vote of tokens.

Section 2.02 Changes to the Authorized Protocol. At any time, an affirmative **majority vote** (50% plus one Zen Token) of the total number of Zen Tokens that have been issued on the Community Release Blockchain (such issued total, the "**Existing Tokens**") can authorize a change to the Authorized Protocol. Each Zen Token entitles the owner to one vote. For the avoidance of doubt, Zen Tokens that have not yet been generated are not included in calculating the total of Existing Tokens. In addition, a change to the Authorized Protocol is not necessarily a change to the underlying Licensed Software. Prior to being voted on for adoption, any proposed change to the Authorized Protocol must first be determined to be eligible for such a vote in the first place. The procedure for this is set forth in Section 2.03(c). Votes are tallied in accordance with the procedures described in Section 2.04(d).

a. The following is a non-exhaustive list of items that require a majority vote (50% plus one Zen Token) of Existing Tokens:

i. a change of the number of proposals that can be adopted per scheduled six month Community Vote. Currently, all eligible proposals are voted on, but only one proposal will be adopted, regardless of differences in the types of changes each proposal puts forth. Votes are consolidated into a single vote. **For example**: proposals for a change in block size and a change in the mining algorithm, are all considered and voted on against one another in zero-sum fashion; only one proposal will be adopted);
ii. a change of the requirement that all new Community Release code to have a hard stop to be a valid candidate for a vote (Section 2.03(c)(iii);
iii. a change of maximum 100 million total supply of Zen Tokens to be generated (Section 1.04);
iv. a change of Zen Token supply curve (Section 1.04);

v. a change of the genesis block necessary for the authorized protocol; or
vi. a change to the voting threshold requirements.

Section 2.03 Scheduled Upgrades. Six months following the Community Release Date, and every six months thereafter, Zen Token holders shall vote on the next Community Release version (each voting session, a "**Community Vote**" and each date of a Community Vote, a "**Community Vote Date**"). Each Zen Token entitles the owner to one vote. See "Voting Schedule" in Section 2.04 below.

a. Changes to Community Release. If a majority (50% of Tokens voted plus one Zen Token) of votes cast (but not necessarily a majority of Existing Tokens) support a new Community Release, such version shall be the next version of the Community Release (as of the date determined in the Community Vote); provided, however, that such new Community Release does not contravene the Authorized Protocol. A change to the Community Release that contravenes the Authorized Protocol is considered a breach of the Software License, automatically terminates your right to use the Software License Software and may be considered copyright infringement. For the sake of clarity, in order to effect a change to the Community Release that would require a change to the Authorized Protocol (i.e. because such new Community Release contravenes the Authorized Protocol in place at the time), the voting requirements in Section 2.02 must first be followed to authorize the necessary change to the Authorized Protocol. That is, in advance of the scheduled semiannual Community Vote, the user can initiate an unscheduled vote on a proposal to change the Authorized Protocol pursuant to Section 2.02. Prior to being voted on for adoption, any proposed changes to the Authorized Protocol must first be determined to be eligible for such a vote, as set forth in Section 2.03(c). Votes are tallied in accordance with the procedures described in Section 2.04(d).

b. No Majority. If no Community Release proposal is approved by a majority of votes cast in a Community Vote, a "runoff" Community Vote shall occur, in which the two proposals that received the highest number of affirmative votes cast in the original Community Vote shall be voted on. If a majority (50% of Tokens voted plus one Zen Token) of votes cast in the runoff support a new Community Release proposal, such version shall be the next version of the Community Release (as of the date determined in the Community Vote).

c. Community Release Proposal Requirements. In order to be eligible for a vote, a proposal for a new Community Release must:

i. be on an actual implementation with publicly available code;
ii. have a blockchain which accepts the initial Genesis Block and the blocks later derived from it based on the 'legitimate' technical consensus mechanism as agreed on in the previous vote (i.e., the longest chain). **For example**: in the period of

time between the Genesis Block and the first Community Vote the legitimate mechanism is SHA-3 proof-of-work. If there is a change in proof-of-work based on the vote that occurs in 6 months (EG to SHA-256) than between interval 1 and interval 2 the legitimate mechanism is Sha 256. Therefore the Authorised protocol at the end of 1 year should contain roughly 64,800 blocks of SHA-256 preceded by 64,800 blocks of SHA-3 that terminate/originate with genesis block X;

iii. implement a stop every 7 months blocks. Malkuth (version one of the Community Release) has a stop after 7 months. Version two is elected by the community, but must include a hard stop 7 months later, unless a majority of 50% plus one Zen Token of Existing Tokens votes otherwise;

iv. include a readme document in the 'git commit' (see Section 2.05) explaining the changes in the proposed version of the Community Release. The description of the updates must discuss the material changes and differences, as well as the people responsible for developing and implementing the proposed version.

d. Determining Proposals Eligible for a Vote. In both the scheduled semiannual Community Votes and the unscheduled votes, in order for a proposed Community Release version and/or potential upgrade to the Authorized Protocol to be eligible for a Community Vote, such proposals must first be approved by a vote of 3% of the total Existing Tokens. Such proposals must be submitted by sending a message signed with a public key associated with a Zen balance to Authorized Nodes by initiating a transaction to the voting smart contract. The transaction message shall include a smart contract with Zen Tokens information clearly indicating the proposal that you wish to be voted on, in the form of plain text or a hash of the git commit and URL of the repository. As mentioned in 2.03(c)(iv), in the case of Community Release proposals, the message must include a readme document describing the changes and team responsible. Messages that do not reasonably communicate a clear voting preference shall not be counted. If a proposal did not obtain the aforementioned vote of approval 3% of Existing Tokens, any vote on such ineligible proposal in a Community Vote shall be ignored, and the user will not be able to vote on another proposal during such Community Vote. Proposals relating to votes to change the Authorized outside of the semiannual Community Vote must first be determined to be eligible for a vote as set forth in this Section 2.03(c) no less than 30 days prior to the actual Section 2.02 vote. Votes are tallied in accordance with the procedures described in Section 2.04(d).

e. No Proposals. If no upgrades to the Community Release are proposed for a given Community Vote Date, a "default upgrade" will occur in which the block

version number is incremented by one, the stop date is increased by 7 months, and nothing else changes.

f. No Quorum Requirement. For the avoidance of doubt, no quorum (whether of Tokens or Token holders) shall be required for a Community Vote.

Section 2.04 Mechanism for Proposal Submission and Voting. Users control the development of the Licensed Software by voting on both the technical changes, and the people responsible for implementing and maintaining those changes. Using the Authorized Node software, users can submit a proposal to be voted on by sending a message signed with a public key associated with a Zen balance to Authorized Nodes by initiating a transaction to the voting smart contract. Similarly, using the Authorized Node software, users can vote on a proposal by sending a message signed with a public key associated with a Zen balance to Authorized Nodes by initiating a transaction to the voting smart contract. The transaction message shall include a smart contract with information clearly and unambiguously indicating your vote or the proposal that you wish to be voted on, as the case may be, in the form of plain text or a hash of git commit and URL of the repository being voted on or proposed to be voted on pursuant to 2.03(d). As mentioned in 2.03(c)(iv), in the case of Community Release proposals, the message must include a readme document describing the changes and team responsible. **Messages that do not reasonably communicate a clear voting preference shall not be counted**.

a. One Vote per Period. All eligible proposals are voted on, but only one proposal will be adopted, regardless of differences in the types of changes each proposal puts forth. Votes are consolidated into a single vote. For example: proposals for changes in block size, hashing algorithm, reward size are all considered and voted on against one another in zero-sum fashion; only one proposal will be adopted.
b. Voting Schedule. The following timeline summarizes the process for the voting provisions discussed herein.
    i. Votes shall occur every six months, beginning with six months from the Malkuth release date.
    ii. In weeks 17-18, Community Release proposals must be disclosed publicly, and the community will determine which proposals are eligible to be voted on (via the 3% eligibility vote set forth in Section 2.03(d).
    iii. In weeks 17-18, the community shall vote on those Community Release proposals that received more than 3% or more in the Section 2.03(d) eligibility voting.
    iv. In weeks 21-22, the community shall tally the votes and determine the integrity of the votes.

v. In the event that no Community Release Proposal received an affirmative majority of votes, the community shall conduct a runoff vote in weeks 23-24.
vi. In weeks week 25-26, users download the voted on Community Release that won the vote and upgrades to version as the new Community Release.
vii. After week 26, there is an upgrade to this next version.

c. Tallying Votes. After each voting period closes, votes are tallied in a distributed fashion: users must collect and count the publicly voting information available on the Authorized Nodes. Users should then publish the results of their tallies by broadcasting such information to the Authorized Nodes in the form of a transaction message. Tokens may not be voted more than once. Messages that do not reasonably communicate a clear voting preference shall not be counted.

Section 2.05 Adherence to Additional Principles. Proposed upgrades to the Community Release and/or Authorized Protocol must preserve:

a. the backward and forward compatibility described in Section 1.01;
b. the Circulating Supply Schedule set forth in Section 1.04;
c. the ability of Token holders to vote on future proposed upgrades to the Authorized Protocol and Community Release; and
d. in the case of Community Release upgrades, a six month expiry of such new version.

Section 2.06 No Contentious Forks. Other than pursuant to the conditions and procedures as described in this document, "Hard Forks" are not an acceptable consensus mechanism. Each community member has the right to keep his or her Zen Tokens whole and not split, which may diminish the value of the Zen Tokens.

Section 2.07 Initial Reserved Powers. Notwithstanding the other provisions of this Authorized Protocol, Blockchain Development LTD and its principals may, until July 1st 2020, issue such updates as are, in their judgement, necessary and advisable for the continued reliable operation of the Zen Protocol blockchain. Such updates may be made in addition to any passed via the other mechanisms permitted by this Authorized Protocol.

END OF AUTHORIZED PROTOCOL

## APPENDIX 1

The double SHA-3 hash of the Genesis Block has the hexadecimal encoding
`57b925330faf7d08f1d9799147258bf8fbb6bfea63795c5162221766321215c6`.