CCC: A **First Principles** Self **Governance** System

Imagine with **Misol** (https://x.com/misolcom)

Satoshi Nakamoto set the bar high with **Bitcoin** in 2008 that we now have the term **First Principles** of **Crypto**. Unsurprisingly, the vast majority of blockchain projects afterwards have fallen short of his standards. **CCC** holds the hypothesis that the holder base of purely first principles tokens such as **XEN**, e**VMPX**, and **XONE** are standards of deviations much wiser, and this wisdom can be harnessed through blockchain self **governance** systems.

Centralization of decision making have plagued governance systems throughout history. Instead of electing individuals, what if we could extract the wisdom from the crowd. The **marshmallow experiment** in psychology is a study of **delayed gratification**. Students are offered one marshmallow and are told if they wait to eat it, they will receive a second marshmallow. The children who waited for the second marshmallow had **greater success** in life measured by SAT scores, educational attainment, body mass index (BMI), and others.

XEN pioneered the idea of distributing tokens based on paying a gas fee on the **Ethereum** network while offering additional tokens if you **delay minting**. Over a year has passed since its inception and currently we have a holder base in XEN that has endured arguably more **challenging conditions** than was set by the **marshmallow experiment**.

CCC executes an educational experiment by creating a token with a fixed total number of near **1 billion** and extracting its holder base from **XEN**, **eVMPX** and **XONE**:

1. **Hold XEN, eVMPX or XONE:** From **January 28, 5 pm US Central** to **Feb 25, 5:00 pm US Central**, your average token holdings will be recorded**.**

2. **Mint CCC:** From **March 3, 2024, 5 pm Central** to **March 24, 2024, 5 pm US Central,** you can mint CCC but will not be tradable.

3. **Multiply CCC:** From **March 24, 2024, 5 pm US Central** to **April 7, 2024, 5 pm US Central**, you multiply your CCC allowing total circulating tokens to reach 1 billion.

4. **Vote with CCC**: From **April 7, 2024 5 pm US Central**, you can create polls and vote with your CCC on-chain to help decide the future of CCC.

It's important to note that everything will be based purely on all 17 first principles of crypto as referenced on firstprinciplesofcrypto.com. Landru and Misol are guided by these principles and volunteering to whatever task the CCC community decides through on-chain polls. CCC is an educational project with Landru and Misol serving as contributing researchers.

Dive with **Landru** (https://x.com/Landru319723)

The first principles of crypto provide guidelines to its practitioners that avoid the pitfalls of most typical crypto projects. Many centralized exchanges have been hacked so the first principles promote self-custody. If the rules of a game are being played before the rules are known to everyone, this can result in front-running/alpha-arbitrage, which is why the first principles promote transparency. The list goes on.

Some projects follow first principles during an initial stage or with an initial idea, but the implementation or later promotions by the founder stray from the first principles.

CCC has a simple goal: to enable initial fair holders of the fair launch tokens XEN, eVMPX, and XONE, to vote and collaborate in a fashion immune from changes to the XEN ecosystem that are beginning to lean away from the first principles. The process for token distribution and use is outlined below.

The CCC token tracks timestamped holdings so that it can be used for on-chain voting. A contract will be deployed on Ethereum which allows anyone to make a poll which can be voted on by CCC holders. Voting for the poll will be for a fixed period of time and will use a timestamp for voteweight that is in the past relative to the time the poll is created on the blockchain. After the voting period is over the final results can be queried from the blockchain to determine which selection won the poll.

A vote-tracking-disablement switch will be included in the CCC contract. This allows the vote tracking for a particular address to be disabled or enabled after it is disabled (the msg.sender must be the address that is being switched). The default condition for vote tracking will be enabled for minters but disabled for non-minters since many contracts that are not capable of being concerned with vote tracking (and hence would want it disabled if they are to interface with CCC) are also not capable of disabling it (since they don't know the function to call to disable it). The owner of an address may disable the vote tracking function for that address, which can reduce gas costs for transfer operations involving that address. They may also enable it any time. If the vote weight is queried for an address that had vote weight disabled for some time period, and the query is for a timestamp within that disabled period, the vote weight will be zero.

CCC cannot be minted to any address except an external wallet address. This will be enforced by the minting contract. One key reason for this is that there is a period where the tokens have been minted but cannot be transferred, so all participants simultaneously gain the ability to transfer. Without the provision for external-wallets-only, control of accounts which hold minted CCC could be traded before the CCC itself can be traded, which would give participating non-external addresses (i.e. participating contract addresses) an advantage.

The following procedures and rules intended to be described so precisely that they are bit-accurate so that anyone could code up the implementation and the data uploaded to IPFS and the contract can be precisely validated by the community.

1. An average price determination period (APDP, determines the average price of XEN, eVMPX, and XONE, in ETH) and average holding determination period (AHDP, determines the average amount that was held for each token, for each address), is intended to be 30 days, or 30*24*60*60 seconds, and will have the same target start time Ts, starting block B_s, ending block B_e, ending target time, Te, actual start time Tas and actual end time Tae.
    a. Ts is determined at time of this whitepaper publication, 1706482800 (January 28, 5:00 pm US Central).
    b. B_s is the first block ("block" and "block ID" are used interchangeably in this document) with a timestamp greater than or equal to Ts.
    c. Tas is the actual start timestamp, which is the timestamp of B_s.
    d. Te is Ts + 30*24*60*60. B_e is the last block with a timestamp less than or equal to Te.
    e. Tae is the actual end timestamp, which is the timestamp of B_e
    f. In the case that any XONE is transferred from/to the "XONE owner address", or XOA (i.e. the address holding 500m XONE as of this writing, which is 0xC73Fc08C931Efe3FCE850C09278472e8a81c2e05), If the APDP and AHDP have not started yet, then Xone average holding amount is instead calculated as a snapshot that is instantaneous at the time of the transaction right before the transfer transaction ("transfer-containing-owner-address transaction", or TCOAT) containing the XOA. Similarly the market cap is the snapshot price, in the most recent swap of the xone-eth 1% liquidity pool, prior to the TCOAT. If the APDP and AHDP have started and block containing TCOAT, call it B_TCOAT, is greater than or equal to 1 + B_s, then the average holding amount and average market cap for xone are computed as if B_e is B_TCOAT except that all transactions at or after TCOAT are treated as not present in B_TCOAT (i.e. calculations do not include transactions after TCOAT, nor do they include TCOAT). In fact the ignoring of certain transactions within B_TCOAT is unnecessary to state since, as may be seen in subsequent procedures and calculations, such transactions would have zero duration, since they are in the B_e being used for xone, and so don't affect average market cap or average holding amount. The calculations of APDP and AHDP are not affected by the TCOAT.
    g. In post processing of the average holding amount for each address, in the case of tracking average holding amounts, the 0 address is always deleted, of course (it ends up with negative balance and is not a real account). The XOA is also deleted.
2. During the average price determination period (APDP), a time-weighted price is calculated, in ETH, for each of XEN, eVMPX, and XONE, under the following rules. The resulting values of APDP will be loaded into the CCC contract. The average price of the

token (XEN, eVMPX, or XONE) is determined by multiplying the price of each token multiplied by the number of blocks it was at that price, divided by the total blocks of the APDP. Note that only the closing price of a block is used (which provides some protection from flash loans).

a. When processing swap events, if the filter for events is such that the earliest block for the filter is B_s, but no relevant swap events exist in B_s, then the price is not known at B_s which is necessary for the calculations. In that case, to determine the starting price of B_s, some search for the most recent swap that occurred before B_s must be done. One algorithm for this is to find a block about 24 hours before B_s, and use a filter for events from that block up to (and including) B_s-1. The last event returned with that filter is the starting price for block B_s. If no events are returned with that filter, use a filter for events that chooses a starting block 48 hours before B_s, and so-on until a swap event is found. To date very few if any periods for any of the 3 tokens xen/vmpx/xone have gone 24 hours without a swap. If an error is returned due to too many swap events being returned from the chosen filter, increase the start of the filter by e.g. a quarter its distance to B_s, or so, until no such error occurs (unlikely as the limit is typically 10,000 events and the tokens generally do not ever have 10,000 swaps in 24 hours). This all is quite intuitive for manually finding such a transaction but we specify the above details to highlight some of the programming requirements necessary to handle arbitrary levels of swap activity.
b. Note that “block durations”, rather “seconds durations” are used for the inner loops of this calculation as well as the average holding calculation below, for simplicity. Fetching timestamps for each block with standard web3 APIs could be extremely bandwidth intensive since it may require fetching the data for an entire block just to get the timestamp data.
c. The algorithm for determining the average price of a token continues as follows.
d. To calculate the average price of a given token, initialize by setting the most recent price MRP to the initial price IP which is the closing price as of B_s, which is either the price of the last swap emitted from B_s, or the most recent price in a previous block as calculated in step 2a. Set the most recent swap block MRSB to B_s.
e. Initialize sum prices SP to (B_e – B_s)*MRP. This is total blocks in the analysis period multiplied by most recent price.
    i. For each swap event that gives a price of the token, processing events in order from blocks B_s + 1 to B_e -1, inclusive, where the block of the current event being processed is B_c, the price of the token (XEN/eVMPX/XONE) in ETH as priced in the current event is P_c (“price current”),
        1. The price difference PD is P_c – MRP.
        2. Set Sum += (B_e-B_c)*P_c
        3. Set MRP equal to P_c

ii. Finally, the average price is SP / (B_e – B_s), which can be done with integer division (truncation) since the prices are in 18-decimals format.

iii. Note that prices in swaps are encoded as square roots and with other precision issues (e.g. 96-bit decimals) that must be handled properly to generate prices like p in 18 decimal format.

f. The average price avg_xen, avg_vmpx, and avg_xone, for each of XEN, eVMPX, and XONE respectively, are loaded into the CCC contract after APDP but before the first mint (see below).

3. An average holding determination period (AHDP), is the same period as the MCDP.

a. The average token amount of an address ATA(addr) is calculated as the sum of the token balance of the address at the end of each block in the MCDP (every block whose ID is greater than or equal to B_s and less than B_e) divided by the number of blocks counted in MCDP (B_e – B_s).

b. An algorithm to efficiently compute ATA(addr) starts by computing an initial table "Ti(addr)" of all addresses and their holdings as of the end of block B_s. Next, the final table T_f is initialized such that for every address the balance "T_f(addr)" is set to Ti(addr)*(B_e – B_s). Then all transfer events are processed in order, where current block B_c increments from B_s + 1 inclusive, to B_e – 1, inclusive. For every transfer, with "from" address A_f, and "to" address "A_t", with "amount" amt:

i. T_f[A_f] -= amt*(B_e-B_c)

ii. T_f[A_t] += amt*(B_e-B_c)

c. After processing all B_c, the next step is:

i. For every entry in T_f, divide by B_e-B_s with ordinary integer division (truncation rounding). Since token amounts are in 18-zeros for all the tokens, XEN, eVMPX, and XONE, such rounding is reasonable and congruent with solidity/blockchain math.

d. For XONE, see previous discussion regarding special cases regarding XOA and TCOAT.

4. Chunking of the snapshot data:

a. Name each of the final T_f tables T_fxen, T_fvmpx, and T_fxone for each of the tokens.

b. For each address that has any positive amount in T_fxen, T_vmpx, or T_fxone (not the 0 address which should be negative, and excluding XOA as previously mentioned), a 512-bit entry (64 bytes, for simplicity) is constructed with 160-bits of address, followed by 96-bits covering the XONE amount, 128-bits covering the xen amount, and 128-bits covering the eVMPX amount, and this entry is appended to the master holding list MHL. The list is sorted by address in increasing order and the list is divided up into chunks of X kilobytes (a power of 2 bytes, making easy compatibility with Merkle trees; probably X will be 2048 for 2, thus holding in this example 32768 entries). The last chunk, if it is not of a full chunk size is padded at the end to reach the full chunk size, with incrementing addresses up to the highest address (0xff..ffff) and 0 amounts for the tokens in those entries.

   i. E.g. if chunks are 32768 entries but there are only 32767 real addresses in the chunk, then one is appended for address 0xffffffffffffffffffffffffffffffffffffffff, with zero for each token amount. If there was by some ridiculous chance a real address with tokens at address 0xff.ffff, then padding would be at second-to-last address 0xff..fffe and zero for token amounts, if there was no real address with tokens at that address. In this way handling discussion of the meaning of empty entries is unnecessary.

c. The data is uploaded, e.g. to IPFS, and a link to the data of each chunk as well as the starting address of each chunk is stored in the CCC contract so that it can be read by users of the contract.

d. The Merkle tree root of each chunk is stored in the contract along with the link and starting address of each chunk (where leaves are hashed using keccak256, and non-leaf nodes are the keccak256 of the concatenated child nodes).

5. The sum token amounts stored in the chunks, sum_xen, sum_vmpx, and sum_xone for each token XEN/eVMPX/XONE respectively, is totaled and stored in the CCC contract.
6. The time period for performing the first mint transaction will be at Ts + 37*24*60*60 (37 days after Ts, one week after integration period completes, with a duration of 30 days. During this time people with wallets with token amounts that have been captured in the chunk data and have merkle roots loaded into the contract can mint a number of CCC roughly equivalent to mint1_CCC, which is derived in the following manner (subject to rounding errors inherent in computers) to target a max supply of 1B tokens (if every outstanding token minted).

   a. Total average ETH value before scaling TAEVBS = (avg_xen * sum_xen + avg_vmpx*sum_vmpx + avg_xone*sum_xone)

      i. Each is stored in 18-zeros format and fixed point math is done as normal in solidity.

   b. During first mint, a user with a given address, has its XONE, XEN, and eVMPX amounts (amt_xone, amt_xen, and amt_vmpx respectively) loaded from the relevant chunk (e.g. by using links stored in the contract, and loading from ipfs). The amount of CCC minted in the first mint CCC1[addr] is:

      i. 1,000,000,000 * (amt_xone*avg_xone + amt_xen*avg_xen + amt_vmpx*avg_vmpx) / TAEVBS

      ii. The total amount of CCC minted in the first mint TCCC1 is tracked

      iii. Note the attempted target supply of approximately 1 billion if every address in every chunk performs a first

   c. The mint transaction receives the merkle proof and the token amounts. The address is the msg.sender which must also be the tx.origin (i.e. external wallet address, as previously discussed).

7. The time period for completing the second transaction starts at Ts + 67*24*60*60 and continues for 14 days.

   a. The additional mint for an address, CCC2[addr] is the following formula:
   b. CCC2[addr] = CCC1[addr] * (1,000,000,000 – TCC1) / TCCC1
   c. Therefore total minted for an address is CCC1[addr] + CCC2[addr]
   d. This second mint gets the final supply much closer to the targeted 1 billion supply because people who minted in the first phase are more likely to mint in the second phase and if they all do perform this second mint, the final supply is very close to 1 billion with some small rounding errors.

8. After Ts + 81*24*60*60, CCC becomes transferable and approvable.

Cat Church LLC & CCC Token **Disclaimer**

No further development by members of Cat Church LLC will be performed in their capacity as founders, after the token launch. The founders will be equal members of the community at that time and all members will be able to shape the use of the token as equals after the launch. There should be no expectation that any managers or founders will develop additional uses cases beyond the launch use cases after launch. No such commitment or promises are being made.

The founders are not selling CCC, it is free to mint, subject to the normal transaction fees on the Ethereum blockchain. The founders do not promote purchasing the token on the open market if the token is made available by some owners for sale. The founders do not believe the token will appreciate and it should not be considered an investment.

This whitepaper is intended for informational purposes only and does not constitute a guarantee of the future performance or value of CCC. The contents of this document are not to be construed as legal, business, investment, or tax advice. Potential token holders are advised to conduct their own due diligence and consult with professional advisors for any legal, tax, accounting, or investment advice.

The issuance of CCC is intended to be in compliance with applicable laws and regulations. However, the regulatory status of digital tokens is subject to significant regulatory uncertainty and may change. Potential token holders are advised to familiarize themselves with relevant legal and regulatory constraints in their own countries of residence.

Investing in digital tokens involves a high degree of risk, including but not limited to market volatility, regulatory changes, and technology risks. Potential investors should be prepared to sustain a total loss of their investment.

While every effort has been made to ensure that the information set forth in this document is accurate as of the date hereof, Cat Church LLC makes no warranties or representations as to the accuracy, reliability, or completeness of this document and disclaims any liability for any errors or omissions.

This whitepaper does not constitute an offer, solicitation, or sale of CCC in any jurisdiction where such offer, solicitation, or sale would be unlawful under the securities laws of such jurisdiction.

Residents of certain jurisdictions may not be eligible to purchase or hold CCC due to legal restrictions. It is the responsibility of potential token holders to ensure compliance with their local laws and regulations.

This document may contain forward-looking statements which are based on Cat Church LLC's current expectations and projections about future events. These forward-looking statements are subject to risks, uncertainties, and assumptions about CCC, and there are important factors that could cause actual outcomes to differ materially from those expressed or implied by such statements.

Cat Church LLC reserves the right to modify, amend, or update this document and its contents at any time and without prior notice. The most current version of this whitepaper, as made available by Cat Church LLC, will supersede all previous versions of this document.

All intellectual property rights in and to the whitepaper, CCC.meme, and all content and logos contained therein are the property of Cat Church LLC. Unauthorized copying, distribution, or use of any part of this document is strictly prohibited.