ARTONOMOUS MECHANISMS DEFINITIONS

MATT BARLIN, MARKUS KOCH, MICHAEL ZARGHAM

Formal definitions of the Artonomous economy and component mechanisms for consideration and Review by Architect and Principal Engineer, Simon De La Rouviere.

1. Introduction

System definition is necessary to design properly engineered solutions. The Artonomous system is designed in terms of its state variables on the blockchain. These state variables represent a certain aggregation of tokens and at times their associated addresses, under conditions and set definitions. Specifically, five state variables in Artonomous are Gallery, Pool, Supply, Votes, and Candidates. Complex systems require definition in order to break the system into solvable components with their respectively well-defined boundaries. We define Artonomous in terms of its mechanisms and roles. Mechanisms define the action space available in the Artonomous system: Art Generation, Art Sale, Patron Bonding, Patron Withdrawal, Patron Staking, Patron Unstaking, and Generator Proposal. Roles are defined as sets of Ethereum addresses from which user actions are taken or from where Artonomous derives information. The role definitions will also serve to provide operational requirements of the mechanisms. Roles in the Artonomous ecosystem are: Caller, Collector, Patron, Voter, Generator, and Developer.

2. System Requirements

System Requirement 1. *Artonomous operates in an entirely decentralized manner.*

Every piece of software needed for Artonomous to work is decentralized. There is no central actor capable of terminating Artonomous or determining how it evolves. Centralized on- and off-chain agents can interact with Artonomous but must not be able to have full control over it.

System Requirement 2. *Artonomous must create and put for sale a piece of art periodically.*

System Requirement 3. *Art pieces are immutable*

Every art piece is represented as a ERC-721 token containing the information necessary for it to be rendered. These pieces of information are immutable, as must be any external content they may refer to.

System Requirement 4. *The Artonomous art generation process can be improved upon*

It must be possible to upgrade the component of the Artonomous system responsible for the visual appearance of art pieces. At the same time, previous versions of such component must be kept unmodified.

System Requirement 5. *Art pieces are visually unique*

Incentives should be such that it would be costly to create copies of previously generated art pieces.

3. System State Variable Definitions

As simple preliminaries define the Ethereum network block height to be indexed by k, the set of all Ethereum addresses to be denoted $\mathcal{A}$, the address of the Artonomous contract by $\alpha \in \mathcal{A}$ and that all other Ethereum addresses may be referenced as some $a \in \mathcal{A}$.

Definition 1. *Define the **Gallery** to be the set of unique art pieces owned by the Artonomous Contract. The set will be denoted $\mathcal{G}$ for Gallery and each unique art piece in $w \in \mathcal{G}$ is a non-fungible token whose state is stored by the Artonomous contract. The set of all works is given by $\mathcal{W}$.*

System Requirement 6. *State System Requirement: The piece of art is represented as an ERC721-token, tradable and transferable on the Ethereum blockchain*

For any work $w \in \mathcal{W}$, there is mapping

$$\text{Owner} : \mathcal{W} \to \mathcal{A} \tag{1}$$

that is to say, that for any $w \in \mathcal{W}$ there is an

$$a(w) := \text{Owner}(w) \in \mathcal{A} \tag{2}$$

is the address that owns the art piece w. Furthermore, the Gallery is formally the set

$$\mathcal{G} = \{w \in \mathcal{W} | a(w) = \alpha\} \subseteq \mathcal{W}. \tag{3}$$

Definition 2. *Define the **Pool** to be the sum of ether owned by the Artonomous Contract, with address $a = \alpha$. The ether belonging to any account a is E_a then the Pool is $E_\alpha(k)$ at block height k.*

The Pool is a subset of E, the quantity of ether in existence

$$E_\alpha(k) \leq \sum_{a \in \mathcal{A}(k)} E_a(k) = E. \tag{4}$$

The quantity E_α is an explicit state of the Artonomous economy, but the system need not concern itself with the distribution of all other ether $E_a(k)$, which keeps the system cardinality bounded.

Further, establish the notation that E_a is the balance in Ethereum of any account $a \in \mathcal{A}$. The Balance of the Pool is the central state variable of the Artonomous Artist Economy; this value serves as the central coupling variable interrelating the component mechanisms presented in the following section.

Definition 3. *The Artonomous Contract may issue a fungible token called a* ***Soul*** *Token in exchange for Ethereum.* $\mathcal{S}$ *is the supply of Soul Tokens. The balance of Soul owned by any address* a *is given by* $S_a(k)$ *The total Soul floating in the Ethereum network is*

$$S(k) = \sum_{a \in \mathcal{A}} S_a(k)$$

which is a state variable immediately observable from Artonomous contract.

Definition 4. *Define the* ***Candidates*** *to be the set of possible Ethereum addresses that Artonomous can delegate the task of generating an art piece to, denoted* $\mathcal{C}$*.*

The simplest rendering of this definition would be to consider every Ethereum address to be a possible Candidate address or $\mathcal{C} = \mathcal{A}$. The majority of candidate addresses under this thinking would not have the ability to generate art pieces. However, this would not have a negative impact on the system, as there would be no incentives towards staking Soul towards such inept candidates.

System Requirement 7. *Each address in* $\mathcal{C}$ *must be in the set of Ethereum addresses.*

System Requirement 8. *The creation of candidates capable of generating art pieces must be incentivized.*

$\mathcal{C}$ will be the set of staking addresses available to Votes.

Definition 5. *Define the* ***Votes*** *to be the record of the number of Soul* S_{a_c} *staked by an address* a *towards an address* c

$$\mathcal{V} = \{(a, c, S_{a_c}) | a \in \mathcal{A}, c \in \mathcal{C}, \sum_{c \in \mathcal{C}} S_{a_c} \leqslant S_a\}$$

Using Solidity data types, $\mathcal{V}$ can be constructed as

```
(address staker, address stakee, uint stakedSoul)
```

System Requirement 9. ***Votes*** *must be staked to an address.*

System Requirement 10. *There should be no value in staking towards an address incapable of generating art pieces.*

4. System Component Definitions

In this section each of the seven mechanisms which interact as part of the Artonomous economy are defined. Those mechanisms are

(1) Art Generation - Art Piece NFT Minting
(2) Art Sale - Auction
(3) Patron Bonding - Soul Minting
(4) Patron Staking - Soul Staking
(5) Patron Unstaking - Soul Unstaking
(6) Patron Withdrawal - Soul Burning

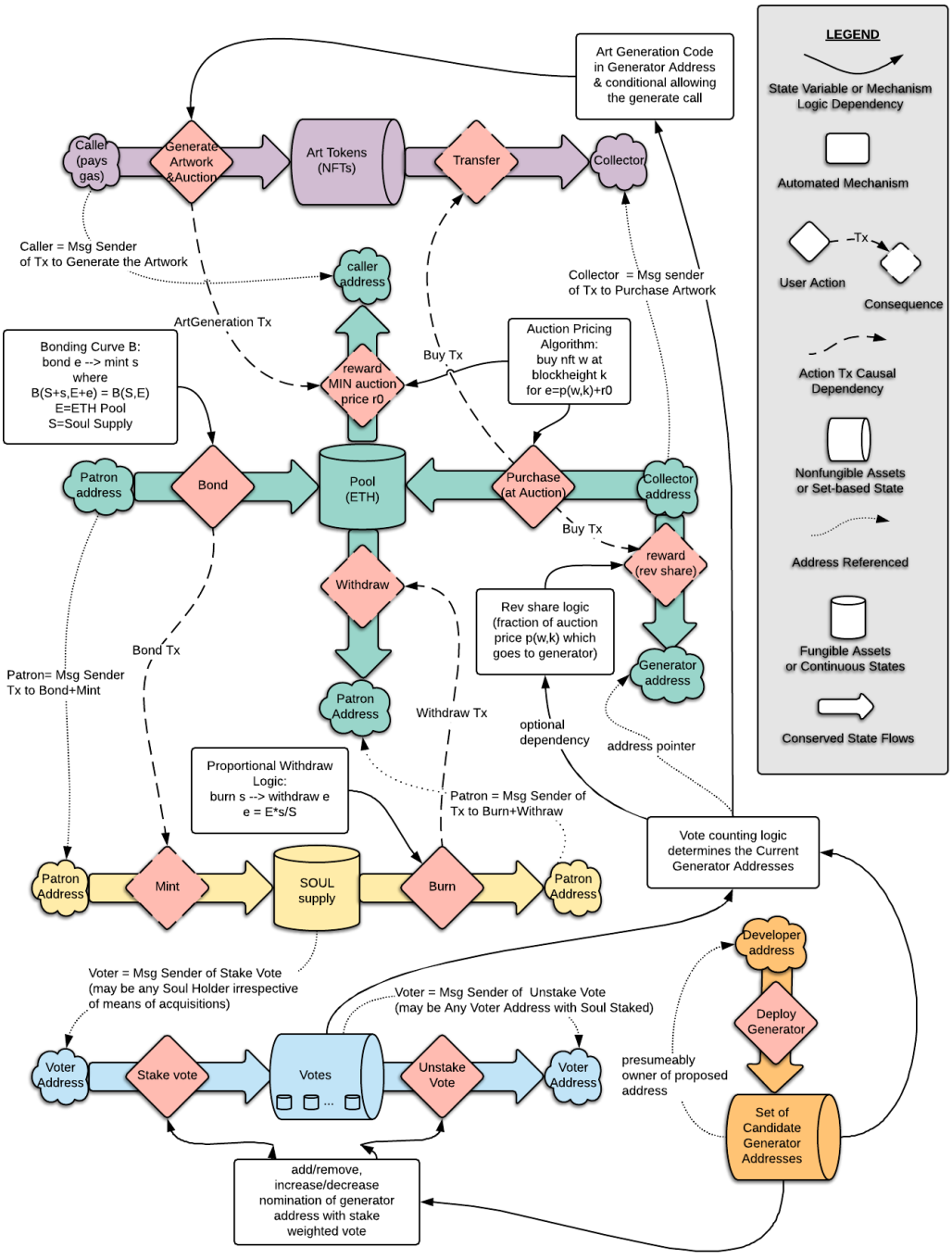

FIGURE 1. System Diagram

(7) Generator Proposal

These mechanisms serve to detail the user action associated with each component. Further development will lead to component requirements for the design of each.

Definition 6. *The **Art Generation** Mechanism is a public method of the Artonomous Contract which triggers the art generation process.*

Due to the passivity of smart contracts, an external actor must trigger the art generation process, at a cost for themselves. Let the generation event of art piece w be $g(w)$ and the block height of $g(w)$ to be $k_0(w)$. The cost of triggering the generation process of art piece w denoted in Ether is defined as $c(w)$.

Once the art generation process is started, Artonomous selects a Candidate G to delegate the art generation to. G will be the generator of art piece w. The selection is made based on the Votes

$$f(\mathcal{V}) : c \in \mathcal{C} \to G \tag{5}$$

Operational Requirement 1. *The art generation mechanism must respect the art generation frequency defined in the system requirements.*

Definition 7. *The **Art Sale** Mechanism is the mathematical inner workings of the public purchase method which allows any user to buy a piece of art piece in the Gallery. Denote the price of purchase of art piece $w \in \mathcal{G}$ as $p_w(\kappa)$ where $\kappa = k - k_0(w)$ represents the number of blocks w has been in the Gallery for.*

Any acceptable **Art Sale** Mechanism must have the form

$$\text{Price} : \mathcal{W} \times \mathbb{W} \to \mathbb{R}_+ \tag{6}$$

where $\mathbb{W} = \{0, 1, 2, ...\}$ is the set of whole numbers and $\mathbb{R}_+$ is nonnegative real numbers. Modeling the output as a nonnegative real is sufficient for modeling purposed but the implementation will in fact require an Integer data type consistent with the Ether cryptocurrency.

The authors propose the following construction:

- Define a global parameter $\gamma \in (0, 1)$ to be the exponential discounting rate
- Set a local variable p_0 to be equal to the largest purchase price that occurred in the last week (approximated by block range)
- At the generation event define

$$p_w(0) = p_0$$

- For each subsequent block decrement

$$p_w(\kappa) = \gamma \cdot p_w(\kappa - 1)$$

Note that this method is consistent with the inexact backtracking line search, a common tool in convex optimization. Implementation notes: the value γ must be stored as two integers η and β such that $\gamma = \frac{\eta}{\beta}$ and $\eta < \beta$ each decrement is subject to rounding errors caused by rounding errors in approximating floating point math with integers.

The event where an art piece w is purchased occurs at block $k_1(w)$, and therefore yields revenue according to

$$E_\alpha^+ = E_\alpha + p_w \left(k_1(w) - k_0(w)\right) \tag{7}$$

for art piece w, where $p_w \left(k_1(w) - k_0(w)\right)$ is computed at the time transaction itself is computed.

The goal of this mechanism is to provide collectors a good opportunity to purchase the art piece at their bid price without requiring extensive transactions. The value p_0 is created at the generation event and the Artonomous can always compute the price associated with a purchase event as part of resolving the transaction. Cite the Cryptokitties auction contract which uses a linear declining price auction with similar mechanics.

Definition 8. *The **Bonding** Mechanism is the mathematical inner workings of the public method which mints Soul in exchange for Ether.*

This mechanism may be implemented as a bonding curve which is a special case of a value function as defined in [1]. Define the bonding function as $B(E_\alpha, S)$.

A simple separable construction would be

$$B(E_\alpha, S) = \frac{g(E_\alpha)}{f(S)}. \tag{8}$$

Interpreting $B(E_\alpha, S) = c$ as invariant the Bonding Mechanism must attempt to enforce,

$$f(S) = c \cdot g(E_\alpha), \tag{9}$$

It suffices to set $c = 1$ as any desired constants can be encoded in $f(\cdot)$ and $g(\cdot)$. A bonding curve is generally expressed with $f(S) = S$ so that it can be interpreted as

$$S = g(E_\alpha) \tag{10}$$

which implies an instantaneous price

$$\frac{\partial B(e, S)}{\partial e} = \frac{\partial g(e)}{\partial e} \tag{11}$$

Such an invariant based derivation means that the designer may choose any bonding curve with $f(S) = S$ and can reasonably expect to have a simply computable mechanism. More complex mechanisms may be designed for general $f(S)$ but the calculus may become difficult and in many cases impractical for derivation or computation.

Having defined the Bonding mechanism as state dependent, it can be viewed as a state update

$$E_\alpha^+ = E_\alpha + e \tag{12}$$

$$S^+ = S + s(e, E_\alpha, S) \tag{13}$$

$$E_a^+ = E_a - e \tag{14}$$

$$S_a^+ = S_a + s(e, E_\alpha, S) \tag{15}$$

(16)

for each user action characterized by an address a Bonding an amount of Ether e and receiving an amount of soul

$$s(e, E_\alpha, S) \tag{17}$$

Definition 9. *The **Patron Staking** Mechanism is the mathematical inner workings of the public methods by which a user locks Soul in the Artonomous contract as a way of demonstrating support for a Generator.*

Definition 10. *The **Patron Unstaking** Mechanism is the mathematical inner workings of the public methods by which a user unlocks Soul in the Artonomous contract as a way of withdrawing support for a Generator.*

Holders of Soul tokens are responsible for determining which of the Candidates will be used to generate Artonomous art pieces. This is done by locking Soul in a smart contract, effectively adding an element to the set $\mathcal{V}$. Conversely, when a Soul holder chooses to withdraw their support for a certain Candidate, they unlock Soul tokens, effectively removing the corresponding element in the set $\mathcal{V}$.

Let a be a Patron currently not staking any Soul for candidate c. The state change equation that represents the staking of S_{a_c} Soul tokens towards candidate c is

$$\mathcal{V}^+ = \mathcal{V} \cup \{(a, c, S_{a_c})\} \tag{18}$$

Let a be a Patron currently staking S_{a_c} for candidate c. The state change equation that represents the withdrawal of their support for that candidate is

$$\mathcal{V}^+ = \mathcal{V} - \{(a, c, S_{a_c})\} \tag{19}$$

Operational Requirement 2. *Require bi-flow symmetry between staking and unstaking actions.*

Definition 11. *The **Patron Withdrawal** Mechanism is the mathematical inner workings of the public method by which users send s Soul to the Artonomous contract in exchange for Ether.*

When an address a requests the withdraw of Ether stored in the Artonomous Smart contract in exchange for s Soul, those tokens are destroyed and Artonomous returns Ether to the sender

$$e(s, E_\alpha, S) \tag{20}$$

resulting in the following state change equations

$$S^+ = S - s \tag{21}$$

$$E_\alpha^+ = E_\alpha - e(s, E_\alpha, S) \tag{22}$$

$$S_a^+ = S_a - s \tag{23}$$

$$E_a^+ = E_a + e(s, E_\alpha, S) \tag{24}$$

Definition 12. *The **Generator Proposal** mechanism is the public method by which a user adds a candidate c to the set $\mathcal{C}$.*

A developer would call the **Generator Proposal** method presumably, but not necessarily, for its proposed address. The state change equation associated with this mechanism is

$$\mathcal{C}^+ = \mathcal{C} \cup \{c\} \tag{25}$$

If the system is designed in such a way that $\mathcal{C} = \mathcal{A}$, this mechanism will be absent.

5. Role Definitions

Within the system, roles are defined as sets of Ethereum addresses from which user actions are taken or from where Artonomous derives information.

(1) Caller
(2) Collector
(3) Patron
(4) Voter
(5) Developer
(6) Generator

Definition 13. *The **Caller** role is characterized by triggering the Art Generation Mechanism, generating the art piece from c and initiating the art auction of that piece w. A caller is the message sender to the Art Generation method in the Artonomous smart contract.*

The Caller will spend gas to create this art piece and begin its auction. Design of this system will influence what class of users will likely be performing this role. An altruistic patron wishing to continue art generation could choose to call. The Developer (see definition below) would have a natural incentive to call if they were to be rewarded with a revenue share on the art it created.

Definition 14. *The **Collector** role is characterized by buying, holding, selling, or trading art pieces w. A collector a is assumed to have some private valuation $v_a(w)$ denominated in Ether for all art pieces w. The Collector will decide to buy, sell, or trade art piece when an opportunity presents itself.*

Definition 15. *The **Patron** role is characterized by bonding, holding, trading or burning Soul. A Patron a is assumed to have some private perceived valuation of Soul V_a denominated in Ether and some quantity of Soul S_a.*

Patrons will be the set of accounts holding Soul tokens. Patrons can be classified as active or inactive. An inactive Patron is simply holding Soul tokens with no action on the Artonomous contract. A patron can be classified as active if it satisfies the condition that the patron is the message sender to Bond, or the patron is the message sender to Burn, or the patron is in the set of Voters.

Definition 16. *The **Voter** role is characterized by locking Soul as a way of demonstrating support for a candidate c. Any Patron a is a Voter at block k if they have some quantity $S_{a_c} > 0$ of Soul locked in the Artonomous smart contract at a block height k and staked towards a candidate c.*

Operational Requirement 3. *Votes are counted over a voting interval*

In order to incentivize Patrons to keep their Soul tokens locked, votes are tallied over an interval of blocks, and not just at the Art Generation block.

Definition 17. *The **Developer** role is characterized by proposing the inclusion in the set of candidates $\mathcal{C}$ of an address $a \in \mathcal{A}$.*

It is proposed that $\mathcal{C} = \mathcal{A}$, and the proposal mechanism be reduced to the act of deploying to the Ethereum network a smart contract that implements a standard interface containing all the information necessary for one to visually render the art piece from a seed.

Definition 18. *The **Generator** role is characterized by pointing to a piece of software (the Generator Software) that creates art pieces.*

For an address c of the set $\mathcal{C}$ to be a Generator, it is necessary, but not sufficient, for there to have been some quantity $S_{a_c} > 0$ of Soul locked towards it in the Artonomous smart contract over a period of time. Other constraints must be implemented in order to meet the system requirements.

Operational Requirement 4. *The Generator Software takes one input.*

Operational Requirement 5. *The art piece created by the Generator Software is deterministic.*

Operational Requirement 6. *The Generator Software runs completely on the client side.*

References

[1] Michael Zargham, Zixuan Zhang, Victor Preciado *A State-Space Modeling Framework for Engineering Blockchain-Enabled Economic Systems*, `https://github.com/BlockScience/artonomous/blob/master/token_engineering/Reference/ICCS2018.pdf` 2018.