

# The Constellation Token Model

Constellation Labs

June 9 2018

## Abstract

We propose the Constellation Token Model, a distributed consensus protocol governed by model that achieves equilibria irrespective of speculative market fluctuations.

## Introduction

A Generative Economy is a living economy that is designed to generate the conditions for life to thrive, an economy with a built-in tendency to be socially fair and ecologically sustainable <sup>1</sup>. This definition could be paraphrased as essentially a game where the rules of play enforce certain invariants, the result of which is emergent stabilization. Essentially, the act of playing the game itself is what enforces the invariant. Distributed consensus protocols, and the currencies they support, can be modeled as generative economies. Network stability is governed by eventually consistent equilibrium and equilibrium is defined in terms of invariant measures (we want to enforce that something doesn't change, or can be relied upon) that are maintained. Why do we need this? It allows for the the enforcement of real world performance specifics that are decoupled from external market dynamics (gas/exchange prices) which allows for technology focused adoption while also providing an external speculation market similar to traditional commodities markets.

## Constellation Economics

Consensus protocols exist to provide a utility and that utility is the basis of a generative economic game. In Constellation, that utility is throughput. Like all

---

<sup>1</sup>Kelly, Marjorie, "Toward A Generative Economy"  
<https://www.opendemocracy.net/ourkingdom/marjorie-kelly/toward-generative-economy>

consensus protocols, access to rewards is governed by delegate selection. In turn delegate selection is governed by a seeder/leacher ratio of transactions validated vs transactions submitted. We will use the term 'utility ratio' to describe the throughput a potential delegate has provided vs the throughput used. As nodes play they get ranked based on their utility ratio. Based on ranks they get access to rewards. Fees are charged relative to the scarcity of the utility that the protocol provides and are only charged if an account breaches its allotted utility ratio. The \$DAG token acts as an inflation mechanism like cash, which can be used to pay transaction fees when an account has not or can not maintain their required utility ratio. \$DAG is injected into the pool of validator rewards, and there is incentive to validate transactions that have an accompanying fee attached. Holders of the token assume the role of central bank, deploying \$DAG capital with the understanding that the free market dynamics of the protocol will ensure their account gets access to the network utility when capital is deployed.

Reputation is a unit that represents a 'stake' of total network utility (throughput). It is a perishable good in the constellation economy as accounts that leave the network will watch their reputation recede until 0 and they are replaced by a new player. Accounts' reputation is varying and falls according to various tiers which follow our definition of rank along with a branching factor.

## Equilibrium and Market Dynamics

Constellation's value can be thought of as stake within a total computational space. Constellation's data model was designed to follow combinatorial models in distributed computing, where network state can be described in terms of symplectic geometry and transitions are defined in terms of a discrete gradient. The value of our network derives from the total computational space available for verifying transactions i.e. its maximum throughput. We can loosely model the total computational space as the volume of 'space' across nodes with respect to rank  $r$

$$\int_M d_1 \wedge \cdots \wedge d_r$$

where  $M$  is a manifold and  $r$  is the maximal rank. Within our model this would be explicitly defined as the numerical integration over the vector space of our cohomological definition<sup>2</sup>

$$\sum_{1 \dots r} \Omega^1 \oplus \cdots \oplus \Omega^r$$

---

<sup>2</sup><https://arxiv.org/abs/1805.07047>

where each  $\Omega$  is a graded abelian group corresponding to a rank in the differential form above. In general, this space is defined in terms of homological integration <sup>3</sup> (over currents) and can be calculated discretely in terms of a product integral, which we will define fully in later work. For our study, we can make a very simple approximation. We define the total computational space as the sum of all available resources  $f(n)$  each node  $n$  provides

$$\Delta(n) = \sum_{i=0}^r e^{\beta f(n)} dn$$

where  $dn$  is a differential over the sorted set of all  $n$  by instance size, such that  $f(n-1) \leq f(n)$ ,  $\beta$  is a branching factor, the maximum number of nodes in a rank and  $f(n)$  is a distribution of computational resources. The relationship between \$DAG and our network's value is that  $\Delta(n) = VR_{\$DAG}$  where  $VR$  represents the rate of validator rewards distribution, a decaying function that will eventually disburse 40% of all total \$DAG, call the \$DAG reserve owned by investors call it supply  $S$ . The  $VR$  is effectively 'locked' into the total space of  $\Delta(n)$  and there is a conversion rate  $C$  that converts \$DAG into some subset of compute space (stake) of  $\Delta$  such that total \$DAG is conserved. Thus, since  $VR - C * S = 0$

$$\Delta(n) = C * S$$

which we can use to relate the influx of value with respect to nodes joining/leaving

$$\frac{\partial \Delta}{\partial n} = C(t) * \frac{\partial S}{\partial n}$$

where  $c$  is our equilibria condition, a 'maximum conversion rate' for spending \$DAG for a subset of  $\Delta$ . When capitol is injected from  $S$  the validator rewards increase by a factor of  $C * s$  where  $s$  is the subset of total supply injected to  $\Delta$ . We can account for the increase in validator rewards with respect to our conversion rate,  $C(t)$  which is equal to our 'minting' curve

$$C(t) = e^{-mt}$$

where  $m$  is the minting rate.

## Analysis and Results

---

<sup>3</sup><https://ncatlab.org/nlab/show/cohomological+integration>