A Credibly Neutral Distribution Framework for Terra Seigniorage

Nicolas Andreoulis, Eui Joon Lee, Nicholas Platias

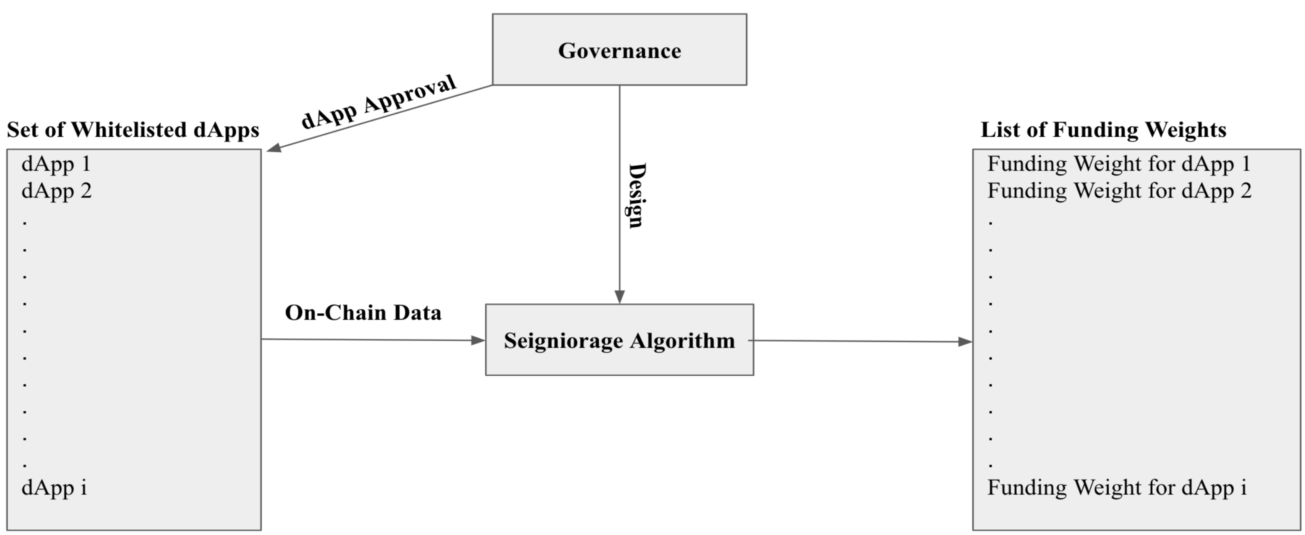
Terra Research

March 2020

Introduction

Until recently, the payment gateway Chai has been the only dApp within the Terra ecosystem. As of March 2020 this changed with KysenPool launching Harvest Wallet, the first DeFi lending platform using Terra. As the Terra ecosystem grows and matures, the use-cases of Terra will diversify with more dApps developing from within the community. A competitive advantage granted by Terra's economics is that dApps can have access to subsidies funded through seigniorage. For such fiscal policy to increase competition and innovation, the seigniorage allocation mechanism should not rely on Terra's foundation discretion, but exhibit "credible neutrality" (Buterin 2020). Such mechanism should 1) be transparent, 2) encourage economic growth, 3) reward economic activity and size, and 4) be hard to manipulate. The goal of this paper is to propose a framework that could fulfil these four objectives.

Allocation Framework in a Nutshell



The above figure summarizes the key structure of the seigniorage allocation framework. As a first step, Luna Holders vote to create a white-list, which is the set of dApps that will compete for seigniorage funds. Luna holders will also vote for the design as well as the parameter choice that will determine the exact allocation algorithm. The voting process should be similar to the existing governance procedure. Then for all white-listed dApps a set of key observable variables will be recorded and stored to be used as inputs for the algorithm. Finally, the output of the algorithm is a set of funding weights, that determine the allocation of seigniorage across different dApps. These funding weights will be updated at a fixed frequency (we suggest a minimum of 14 days), to take into account changes in the economic contribution of each dApp [1].

[1] It's worth noting that this process is activated only when there are resources in the community pool. Accordingly, periods where no new seigniorage has been generated will be skipped.

Variables that determine Seigniorage

For an automated and transparent seigniorage allocation mechanism to succeed, it needs to be calibrated according to observable variables that meet the following criteria.
The variables should be: 1) readily observable on-chain and easy to process, 2) easy to understand 3) adequately capturing economic size, activity and growth, 4) not easily manipulable.

With these criteria in mind, we suggest the following variables:
To capture **economic activity** we are proposing the use of **tax contributions and swap fees** associated with each dApp. The benefit of using tax contribution over transaction volume, is that it's harder to temporarily manipulate it, since taxes are capped at 1 SDT per transaction.
To capture **economic size** we are proposing a measure for **total value locked** (share of assets in the wallets associated with each dApp).
Finally to capture **economic growth** we will be considering the above variables not only in levels but also in **growth rates**.

Tracking Variables Across dApps -TNS

The seigniorage allocation mechanism should be able to accurately track economic variables across a range of different dApps. To do so, we need a way to identify, the key wallets associated with each dApp. The upcoming introduction of Terra Name Services (TNS), will provide a solution to this problem. To be considered by the seigniorage algorithm, all whitelisted dApps will require to register with TNS. With TNS, each dApp, will have a human-readable identifier, that will be used to specify the wallets associated in that network. The tree like structure of TNS, will enable us to separately evaluate each dApps' contribution[2]. An example may be helpful to illustrate the process. According to Figure 1 shows, Chai would register under "chai.terra.money". Accordingly, all merchant wallets would be registered as "X.chai.terra.money". To track chai's contribution, the seigniorage algorithm would then simply evaluate everything within the "chai.terra.money" tree.

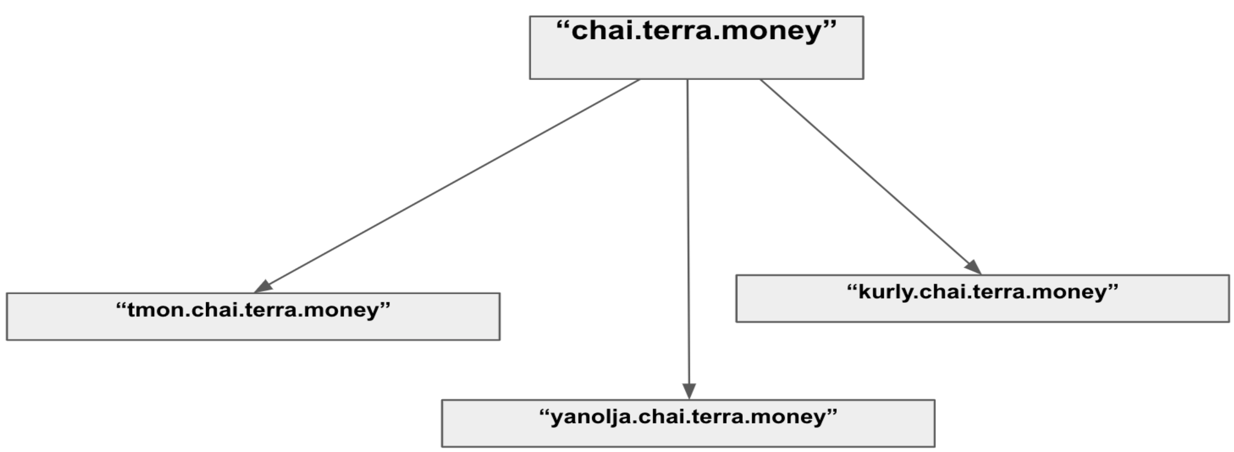


Figure 1: Example of TNS structure.

[2]Explaining in detail the TNS protocol is beyond the score of this paper. The interested reader could read about TNS here.

How to avoid Manipulation?

To avoid manipulation we are considering the following tools.
1) Each dApp has to be white-listed through governance to compete for seigniorage funds. This ensures that dApps developers will have some credibility within the community.
2) All variables will be considered in a long enough window (we suggest at least 14 days) and/or a moving average, such that to minimize the gains from temporary manipulations.
3) Since it's computationally infeasible to collect and store all these data on-chain for each block for an extended period of time, we will collect a random sample of snapshots over time. To determine, which blocks will be sampled, we will use the hash of blocks as a source of randomisation. Obviously the larger a block's header, the smaller the chance of abuse, but the higher the computational cost.
4) We will penalise volatility in observables like total value locked that are easy to be manipulated. By introducing such a penalty, we make short-term manipulation by dApps harder, while also encouraging stable growth.
5) Finally, we will set up a black-list process, that will allow Luna holders to permanently ban dApps from receiving seigniorage funds. This should be considered as a last resort tool, to punish developers who have demonstrably attempted to manipulate the system. Hopefully such mechanism will ensure that the long-term gains of playing by the rules, will exceed the short-term profit of manipulation.

Variable Definition

Given the above discussion, the revenue contribution of for each dApp i, is used as a proxy for economic activity and defined as:

$$Tax_i = \sum_n (TxTax_{i,n} + SwapFees_{i,n}) \tag{1}$$

$TxTax_{i,n}$ refers to the transaction fees contributions to wallets associated with each dApp i and $SwapFees_{i,n}$ refers to swap fees contributions from wallets associated with each dApp i. Revenue contributions Tax_i are defined over a sample of n blocks across a minimum of 14 days.
As a measure of economic size, we propose the use of total value locked defined as:

$$TVL_i = \sum_n f(Balance_{i,n}) \quad \text{where } f' > 0 \text{ and } f'' < 0 \tag{2}$$

where $Balance_{i,n}$ is the sum of all the wallets balances that are associated with a specific dApp i over the same n blocks as in the Tax_i case. $f()$ is a monotonic and concave transformation. This is an intuitive shortcut to penalise volatility. In the absence of it, gaming the algorithm would become much easier. If we were going to simply sum the balances on each dApp, a large temporary liquidity injection would have a dramatic impact on the total value locked, even if it was short-lived. By forcing total value locked to have diminishing returns on balances over time, we ensure that for a given level of total balances, total value locked is maximised, when the variance over time is minimised. However, this decision has the following trade-off, the more concave the transformation we apply (smaller second derivative), the smaller the gains of short-run manipulation, but due to diminishing returns this comes at the cost of failing to reward economic contribution. To make the reasoning concrete we suggest using the intuitive transformation of the square root:

$$f(x) = \sqrt{x} \tag{3}$$

which sufficiently penalizes volatility while also allowing for differences across dApps to have a significant impact on total value locked.

Funding Equation

Using the definition from the previous section we use a simple yet intuitive specification that captures contribution due to economic activity, size, as well as growth. Funding weights due to differences in levels for each dApp i are being determined by:

$$w_{i,level} = \lambda \frac{Tax_i}{\sum_j Tax_j} + (1-\lambda)\frac{TVL_i}{\sum_j TVL_j} \quad \text{where } \lambda \in [0,1] \tag{4}$$

where $\frac{Tax_i}{\sum_j Tax_j}$ is the share of revenues associated with each dApp and $\frac{TVL_i}{\sum_j TVL_j}$ is the share of total value locked. *The key parameter is* λ. As the simulation results demonstrate λ different values of λ affect seigniorage allocation in the following way: **Higher** λ **reward dApps** with **larger economic activity**, measured through revenue contributions. **Lower** λ **reward dApps** with **larger economic size**, measured through total value locked. A subtle implication of the above distinction is that for an identical level of transaction: **Higher** λ values all else equal tend to **reward dApps** that have 1) **relatively smaller average transactions**, since a smaller fraction of transactions will hit the SDT cap, as well as 2) **higher swap activity**, since swaps transactions face a higher tax rate. **Lower** λ values of tend to **reward dApps** with **lower velocity**, since this implies that a higher fraction of transactions is locked within the dApp at any point in time. The above funding weight focuses solely on levels, which biases allocation in favour of incumbent dApps, who tend to be larger. Therefore to take growth into account we define the funding weights due to differences in growth as:

$$w_{i,growth} = \lambda \frac{\Delta Tax_i^*}{\sum_i \Delta Tax_i^*} + (1-\lambda)\frac{\Delta TVL_i^*}{\sum_i \Delta TVL_i^*} \quad \text{where } \lambda \in [0,1] \tag{5}$$

where:

$$\Delta x_i^* = max(0, \frac{1}{13}\sum_{t=1}^{13}\frac{\Delta x_{i,t}}{x_{i,t-1}}) \quad \text{where } x_i = Tax_i \text{ or } TVL_i \tag{6}$$

Equation 6, ensures that funding weights can never be negative. λ has the same trade-offs as before but focusing on rates rather than levels. Then the final funding weights for each dApp are being determined by a weighted average of 4 and 5 defined as:

$$w_i^* = \alpha w_{i,level} + (1-\alpha) w_{i,growth} \quad \text{where } \alpha \in [0,1] \tag{7}$$

The parameter α determines the weighting on levels relative to growth. Lower values of α reward dApps that are growing faster, even if their contribution is smaller. .
To conclude, parameter trade-offs can be summarised in the following matrix:

		λ	
		Economic Size	Economic Activity
α	Growth	low α, low λ	low α, high λ
	Levels	high α, low λ	high α, high λ

1)Higher λ reward dApps with larger economic activity.
2)Lower λ reward dApps with larger economic size.
3)Higher α reward dApps rewards dApps that are already relatively large.
4)Lower α reward dApps rewards dApps that are growing relatively faster.

Simulation Assumptions - Transaction Volume

The purpose of this section is to illustrate the trade-offs associated with different parameter choices by simulating the above algorithm in a reasonable environment. For simplicity we are restricting the set of white-listed dApps to 2. There is a large incumbent payment based dApp and a smaller but faster growing deposit based entrant. The dApps in our exercise resemble in spirit the two existing dApps, but the purpose of this exercise is not to predict how the algorithm will redistribute seigniorage between Chai and Harvest, rather than merely an exploration of the impact of different parameter choices.

We start by postulating the initial conditions and growth rate for each dApp. The incumbent dApp (dApp1) starts with 4 times the transaction volume of the entrant dApp (dApp2), but dApp1 has more than 2.5 times the average daily growth rate of dApp2. We assume that within a year dApp1 doubles its transaction volume, while dApp2 increases its transaction by 7x. To make the above example concrete, while allowing for some natural volatility, we simulate these trajectories 20 times using Brownian Bridge Process and take the median simulation as a baseline. Figure 2 and 3 show the simulated transaction volume for each dApp for the next year.

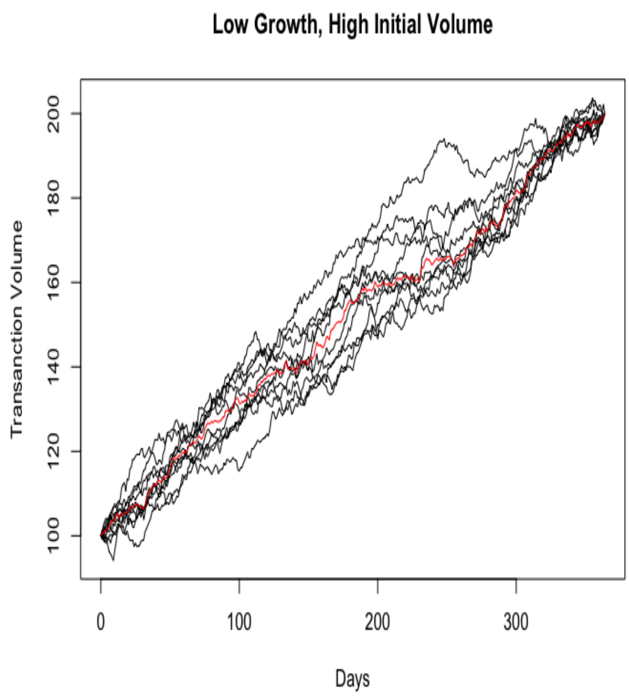


Figure 2: TxVolume for dApp 1

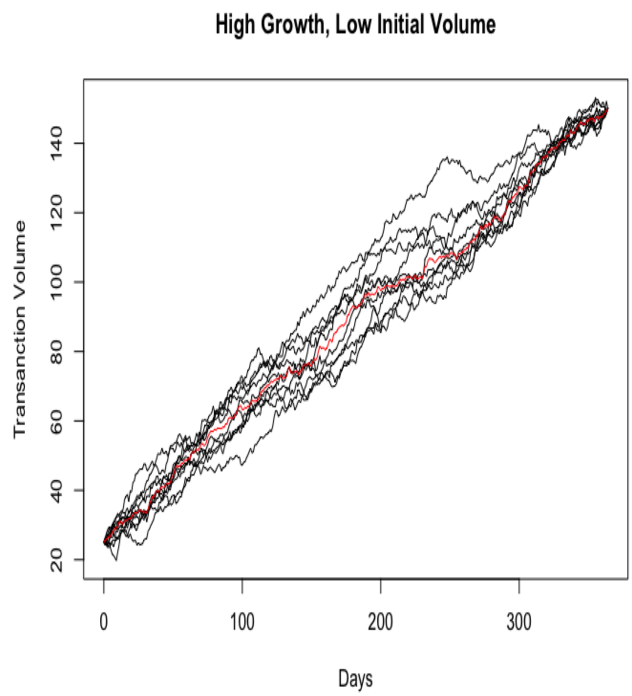


Figure 3: TxVolume for dApp 2

Simulation Assumptions - Revenue Contribution

To reduce dimensionality we express revenue contribution as a linear function of transaction volume. Revenue contribution at day d and dApp i is defined as:

$$Tax_{i,d} = (a\ txrate_{i,d} + (1-a)\ swaprate_{i,d})TxVolume_{i,d} \tag{8}$$

For the payment dApp 1, we assume that all the revenue contribution comes from taxes (ie. $a_{dApp1} = 1$) and that 0.5% of the transaction volume becomes revenue. This implies the with the current tax rate of 0.675% only about 20% of the transactions hit the SDT cap (which is roughly in line with Chai's data). For the smaller deposit dApp2 we believe that average transactions will be larger. We assume that about 60% of the transactions hit the SDT cap. However we also assume that swap fees contribute to 1/3 of the revenue contribution ($a_{dApp2} = 2/3$) with an average rate of 10%. The above assumptions imply that for dApp1, 0.5% of the transaction volume becomes

revenue, while for dApp2, 0.46% of the transaction volume becomes revenue.
The take-away of this exercise is that **for identical transaction volume**, **revenue will be larger** for dApps with 1) **relatively smaller average transactions** as well as 2) **higher swap activity**. The above calibration implies that **identical transaction volume maps** to about **1.09 times larger revenue contribution for dApp1 relative to dApp2**.

Simulation Assumptions - Total Value Locked

We also express total balance associated with each dApp as a linear function of daily transaction volume:

$$Balance_{i,d} = \frac{TxVolume_{i,d} * 365}{Velocity_{i,d}} \tag{9}$$

Specifically, the balance at day d and dApp i is defined as the annualised transaction volume of that dApp i over its annual velocity of money. For dApp2, we assume that people withdraw their money on average 4 times per year, so we set the velocity to be normally distributed with an average value of 4. For the payment dApp1 velocity has two aspects: 1) KRT can be used for payment activities, where we assume a settlement period of 14 days, so we set velocity to be normally distributed with a mean of 365/14=26). KRT can also be used for top-ups, where we assume the same average velocity value as in dApp2. Accordingly. the total velocity for dApp1 is the weighted average of those two cases, where the weights are the share of KRT being used for payments and deposits. We assume that 80% of KRTs are devoted for payments and 20% for deposits. Therefore the final velocity for dApp1 is normally distributed with an average value of $0.8 * 26 + 0.2 * 4 = 21.6$.
The take-away of this exercise is that **for identical transaction volume**, **balance and therefore total value locked will be larger** for dApps with **relatively lower velocity**. The above calibration implies that **identical transaction volume maps** to about **5.4 times larger balance and 2.34 ($\sqrt{5.4}$) larger total value locked for dApp2 relative to dApp1**.

Simulation Assumptions - Summary

The table below summarizes all the assumptions regarding the data generating process of each dApp.

	dApp1	dApp2
Transaction Volume Initial Level	100	25
Transaction Volume Terminal Level	200	150
Transaction Volume Daily CAGR	0.19%	0.49%
Tax Rate - *txrate*	0.5%	0.2%
Swap Rate - *swaprate*	0	1%
Fiscal share of Taxes - a	1	2/3
Fiscal share of Swaps - $(1-a)$	0	1/3
Revenue Rate - $(a\ txrate + (1-a)\ swaprate)$	0.5%	0.46%
Velocity	$\mathcal{N}(21.6,\ 1)$	$\mathcal{N}(4,\ 0.3)$

Table 1: Data Assumptions for the Simulation

Simulation Results

We run each simulation over a period of a year, with the data generating processes explained in the previous sections. The funding weights are being updated every 2 weeks, so a total of 26 times per year. We summarise the information the following way. For each parameter value we compute for the next year the average funding weight for each dApp. Figure 4 summarizes the simulation results for different parameter values.

When $\alpha = 0$, the funding equation corresponds to the growth equation and we only care about economic growth. This means that most of the funding shifts to the faster growing dApp2. Under this scenario the choice of λ becomes insignificant. This is because changes in total value locked and revenue contributions are highly collinear.[3]. Finally it has to be noted that since growth is much more volatile than levels over time, funding weights will vary more over time, the lower the α (see Table 2 for details).

When $\alpha = 1$, the funding equation converges to the level equation and growth rate becomes insignificant. Under this scenario λ becomes much more relevant. Higher values of λ reward economic activity more that size, so resources are being shifted from dApp2 to dApp1.

For the intermediate values of α, the situation is between the two extremes.

To conclude:
The lower the α: 1) the more we focus on growth vs levels (benefiting dApp2 vs dApp1) 2) and the less important the distinction between total value locked and revenue contributions (λ matters less). 3) the higher the volatility of the funding weights over time.
The higher the α: 1) the more we focus on levels vs growth (benefiting dApp1 vs dApp2), 2) the the more important the distinction between total value locked and revenue contributions. 3) the lower the volatility of the funding weights over time.
The lower the λ: 1) the more we reward economic size vs economic activity (benefiting dApp2 vs dApp1).
The higher the λ: 1) the more we reward economic activity vs economic size (benefiting dApp1 vs dApp2).

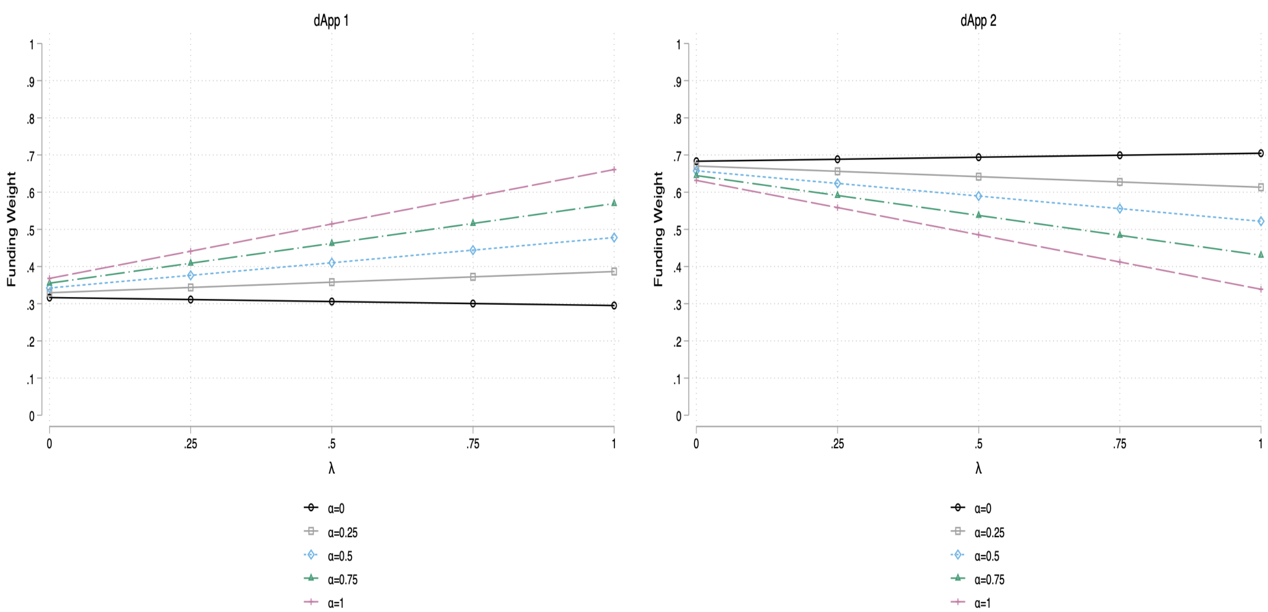


Figure 4: Simulation of Final Funding Weight

[3]In the simulation this by construction, since both total balance and revenue receipts are a linear function of transaction volume. However we expect these two variables to have a strong positive correlation.

Table 2: Standard Deviation and Parameters

SD of Funding Weights	Parameters
.3386711	λ=0, α=0
.2559117	λ=0, α=.25
.1703268	λ=0, α=.5
.0869095	λ=0, α=.75
.0334209	λ=0, α=1
.2576151	λ=.25, α=0
.1920441	λ=.25, α=.25
.1275015	λ=.25, α=.5
.0670272	λ=.25, α=.75
.0398622	λ=.25, α=1
.1749796	λ=.5, α=0
.128919	λ=.5, α=.25
.0849066	λ=.5, α=.5
.0488359	λ=.5, α=.75
.0463413	λ=.5, α=1
.0987366	λ=.75, α=0
.0686174	λ=.75, α=.25
.0432291	λ=.75, α=.5
.0350735	λ=.75, α=.75
.0528444	λ=.75, α=1
.0613087	λ=1, α=0
.0338472	λ=1, α=.25
.0153452	λ=1, α=.5
.0320668	λ=1, α=.75
.0593636	λ=1, α=1

Conclusion

The purpose of this paper was to suggest, a general seigniorage allocation framework that could be implemented on-chain in the near future. The efficacy and robustness of such framework will crucially depend on the parameter choices, that will be determined by Luna holders. As a first step, we identified the main trade-offs associated with various parameter choices. We think that a healthy and transparent governance process would entail revising the simulation results with realized and projected metrics for each dApp. Votes should follow a detailed analysis of that sort so that governance is informed and impact of votes is clear.