Godzillion: A Decentralized Startup Crowdfunding System

Rodrigo Sainz
rodrigo@godzillion.io

Cristóbal Pereira
cristobal@godzillion.io

Eduardo Portugues
eduardo@godzillion.io

www.godzillion.io
August 2017

Abstract. A purely decentralized Smart Contract Architecture that operates on the Ethereum Virtual Machine as a (global) primary and secondary Startups Tokens market place, would allow friction decreases in the Startup issuance process, security increases in the ownership record-keeping, and trades executions exchanging Startup Tokens for value without passing through any intermediary or central server. Blockchain decentralized executions and confirmations provide part of the solution, lowering costs of intermediation and recordkeeping, but the main benefits are lost if a trusted third party is still required to screen startups in order to gain access to an auction process and issue their tokens in the crowdfunding market. We propose a solution to the screening centralized problem by using a DApp Token (GODZ) as an economic voting system reward, giving economic incentives for people to participate in the screening process; and if the voters approved, the Startups can launch crowdfunding campaigns. To be published and attract voters, a Startup creates a Voting Smart Contract and provides it with some amount of GODZ as an economic incentive to the voters instead of paying that fee to a centralized trusted third party. This amount of GODZ will be available to the winning voters (yes or no) as a reward after the auction period ends, whatever the result is. Because the internal rate of return of participants in the voting process depends on the Startup reward and the amount as votes on each side of voting, we expect participants will compete to reach an economic equilibrium in voting pricing.

Godzillion, the state-of-the art decentralized marketplace where people **create, vote and trade Startups**

All rights reserved for mifuturofinanciero.com SpA

Index:

All rights reserved for mifuturofinanciero.com SpA

1. Alternative Assets on Blockchain, why?

One of the most pressing challenges for global investment fund managers is to find ways of achieving higher returns for their investments without giving up liquid trading capability.

To achieve higher yields and low correlations, portfolio diversification has been broadened to include assets that are known as Alternative Assets, where the main categories include Real Estate Funds, Private Equity Funds, Venture Capital Funds and Early Stage Investments.

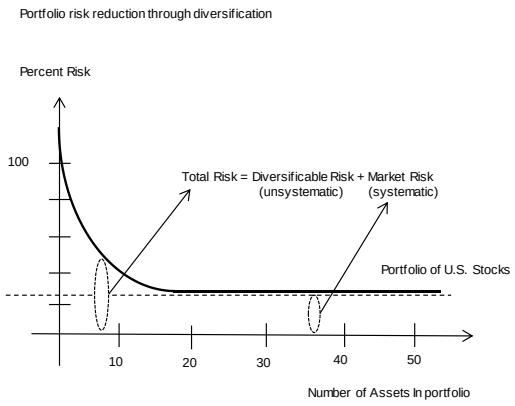
As an example, Cambridge Associates calculated that Early Stage Venture Capital funds returned, on average, over 55% per annum (20-Year).

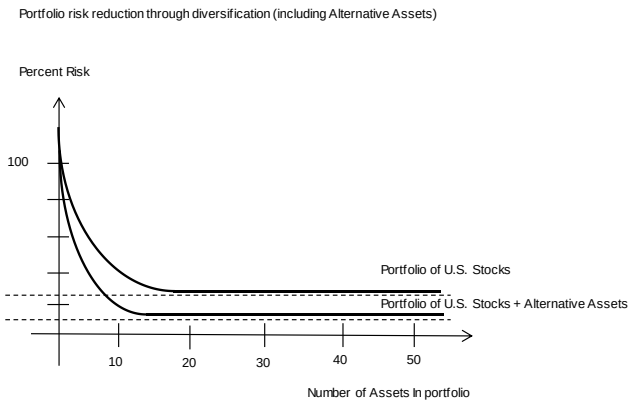
U.S. Venture Capital Index and Selected Benchmark Statistics

Index	1-Quarter	1-Year	3-Year	5-Year	10-Year	15-Year	20-Year	25-Year	30-Year
Cambridge Associates LLC U.S. Venture Capital Index®	**- 3,25**	**4,78**	**19,36**	**14,63**	**10,19**	**4,86**	**30,02**	**25,10**	**18,21**
U.S. Venture Capital - Early Stage Index	**- 3,16**	**5,97**	**20,70**	**15,88**	**10,42**	**3,91**	**55,97**	**34,02**	**22,65**
U.S. Venture Capital - Late & Expansion Stage Index	**- 2,59**	**2,57**	**15,25**	**11,29**	**12,16**	**6,83**	**9,99**	**13,03**	**12,40**
U.S. Venture Capital - Multi-Stage Index1	**- 3,61**	**3,54**	**18,55**	**13,80**	**9,21**	**6,17**	**11,13**	**14,03**	**12,05**
Barclays Government/Credit Bond Index	3,47	1,75	2,42	4,04	4,93	5,03	5,62	6,23	6,57
Dow Jones Industrial Average Index	2,20	2,08	9,29	10,27	7,54	6,55	8,38	10,08	10,73
Dow Jones U.S. Small Cap Index	1,50	- 9,52	7,25	8,21	6,54	9,08	9,25	--	--
Dow Jones U.S. TopCap Index	1,06	0,86	11,58	11,37	7,14	6,23	8,00	--	--
Nasdaq Composite Index*	- 2,75	- 0,63	14,23	11,86	7,61	6,70	7,72	9,69	8,92
Russell 1000® Index	1,17	0,50	11,52	11,35	7,06	6,28	8,11	9,47	9,94
Russell 2000® Index	- 1,52	- 9,76	6,84	7,20	5,26	7,65	7,68	9,29	8,58
S&P 500 Index	1,35	1,78	11,82	11,58	7,01	5,99	7,98	9,28	9,93
Wilshire 5000 Total Market Index	1,18	0,24	11,26	11,01	6,95	6,62	8,04	9,40	9,75

Source: https://www.cambridgeassociates.com/benchmark/u-s-venture-capital-2016-q1/
Data as of March 31, 2016

Portfolio Managers' decisions to include or avoid those assets are not determined by the high risks of the individual assets; instead, their decisions take into account the correlation between Alternative Assets and their classic portfolio assets of listed stocks and bonds.





As returns on Alternative Assets do not fluctuate in synchronization with the rise and fall of industrial country stock and bond markets. Instead their returns are driven by different causative factors. Their returns are uncorrelated with returns of other assets. The low correlation makes it possible to create portfolios that can deliver high returns with low variance.

The formula for the variance of a portfolio's return illustrates this point.

$$\sigma\ of\ the\ portfolio\ expected\ return = \sqrt{w_{US}^2\sigma_{US}^2 + w_{VC}^2\sigma_{VC}^2 + 2\,w_{US}\,w_{VC}\,\rho_{US-VC}\,\sigma_{US}\,\sigma_{VC}}$$

All rights reserved for mifuturofinanciero.com SpA

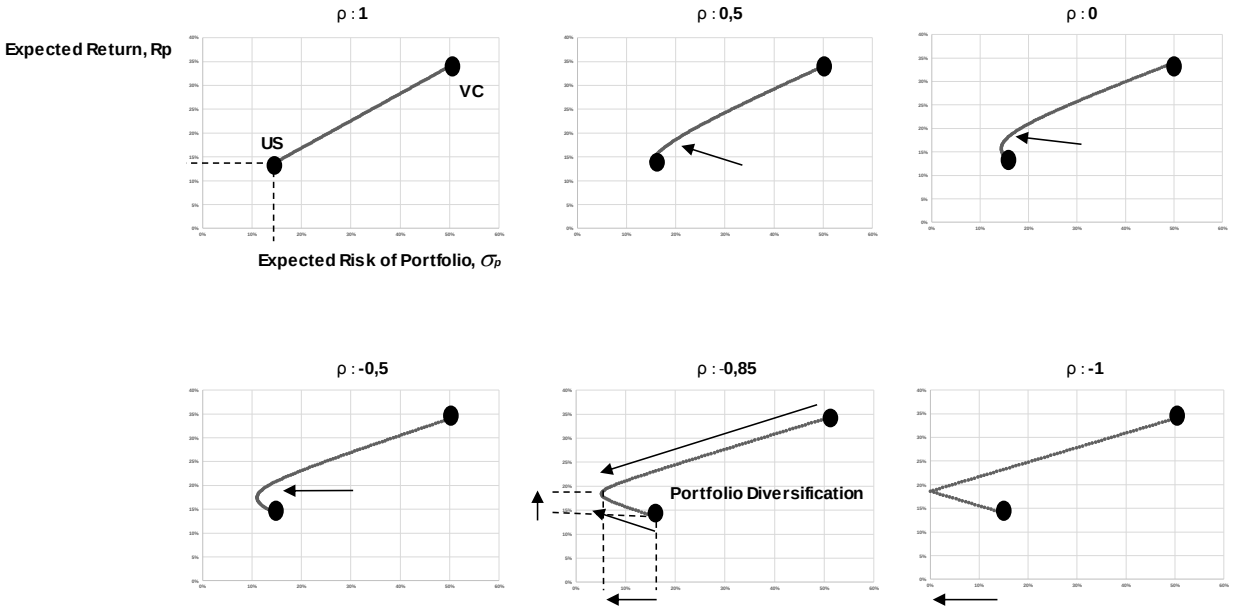
The formula says that the variance (σ) of the expected return on a portfolio with two assets is the weighted average of the returns that its two assets deliver, adjusted for the covariance between the returns on the two assets. If the two assets' returns are highly correlated, then diversification does not lower the variance of the portfolio's return.

But if the two assets' returns are uncorrelated it is possible that the portfolio's returns would be high and stable, even though the returns of the individual assets might be volatile. This approach works well when more than two assets are combined to create a portfolio. With three or more assets, the portfolio's return can be more stable.

We can illustrate the gains from diversification, going from correlation coefficient 100% to perfect inverse correlation, -100%. Main results here, calculations details on the appendix above. Assuming this basis scenario and different correlation coefficients,

	Expected Return	Expected Risk (σ)
United States Equity Index (US)	14%	15%
Venture Capital index (VC)	34%	50%

The Gains from Portfolio Diversification



We can appreciate that combining these two assets, results in an optimization in the risk return payoff. So, as the Alternative Assets have low and sometimes negative correlation with their classic investments, adding Alternative Assets will add value to their portfolio diversification, delivering higher and more stable portfolio returns.

This logic was presented in 1952 by professor and Nobel prize winner Harry Markowitz, and it has been widely put into practice by portfolio managers. Allocating a part of one's portfolio to Alternative Assets is a well-tested way of achieving diversification, while obtaining yields that can be much higher than what is obtainable in stocks and bonds of large companies.

The main problem of this strategy is when the Alternatives Assets has a low or zero liquidity; that puts them into a less appealing category, and overshadows the benefit that they can bring. The lack of liquidity comes from the conventional market structure. There is no aggregated global market place for Alternative Assets.

The issuance and trading processes are awkward and archaic, handled by different centralized trustees and different centralized intermediaries. This structure involves high costs of bookkeeping, custody and auditing, and makes transfers of ownership slow and expensive.

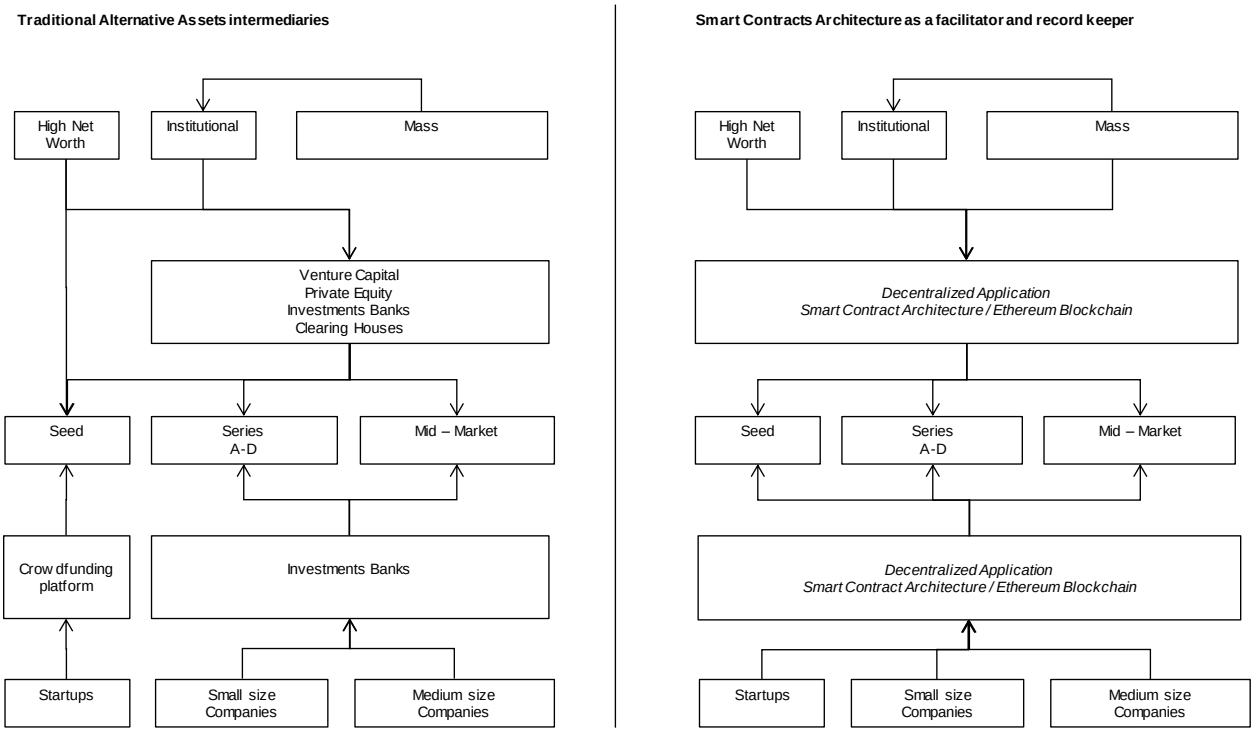
So, with the conventional market structure, Alternative Assets have high fees in the selection, issuing, tracking and auditing processes, and low or non-existent possibilities for trading. Those attributes make them buy-and-hope investments, difficult to value, and hence less suitable for many types of investors.

All rights reserved for mifuturofinanciero.com SpA

All these frictions of the traditional financial system reduce the returns that investors obtain, which affects the value of Alternate Assets in the secondary market, and those frictions make it harder for issuers to place the securities in the primary market. In consequence investors are unable to take advantage of opportunities that would be profitable if those costs were lower and if issuance and trading were more streamlined.

To solve those problems, we have implemented and are operating a decentralized (global) primary and secondary Alternative Assets (seeds, Startups and Small Companies) market place based on Economic Incentives and Smart Contract Architecture, that operates on the Ethereum Virtual Machine. The processes of issuing and trading Tokens that can represents ownership of assets are implemented using a Decentralized Application (DApp). We call it Godzillion.

The Decentralized Application acts as mechanism to facilitate issuance (via crowdvoting and crowdfunding) and trading (via swaps of Tokens). It operates continuously and records all transactions into the blockchain.



The Smart Contract Architecture (Godzillion) operating on top of the Ethereum Blockchain achieve several objectives: it gives economic incentives for people to participate, decreases friction in the issuance process, increases security in the ownership records, and executes trades exchanging Tokens for value without passing through any intermediary or central server.

This solution helps to remedy the drawbacks that have hindered the full acceptance of Alternative Assets in the traditional financial system. It decentralizes the economic incentive of Issuing, and greatly speeds the cumbersome processes of trading and settlement. It achieves its goal with rewards, lower issuance costs and better liquidity to sell and buy Alternative Assets on a decentralized Blockchain secondary market. Its superior design attracts capital to real economy sectors that can offer high returns on investment.

2. The early stage paradigm.

Small companies, in order to raise small amounts of money and start their operations, must pay an upfront fee to an Investment Bank or a Crowdfunding Company to be evaluated. If they passed that first hurdle, they must pay a second fee, as a percentage of the capital raised in the auction process.

Crowdfunding investors support the Startups issuances and buy small blocks of equity for less than 50 USD but after this primary issuance process, there has been no exchange where early stage investors can trade those small blocks of equity with a low cost and efficient settlement procedures.

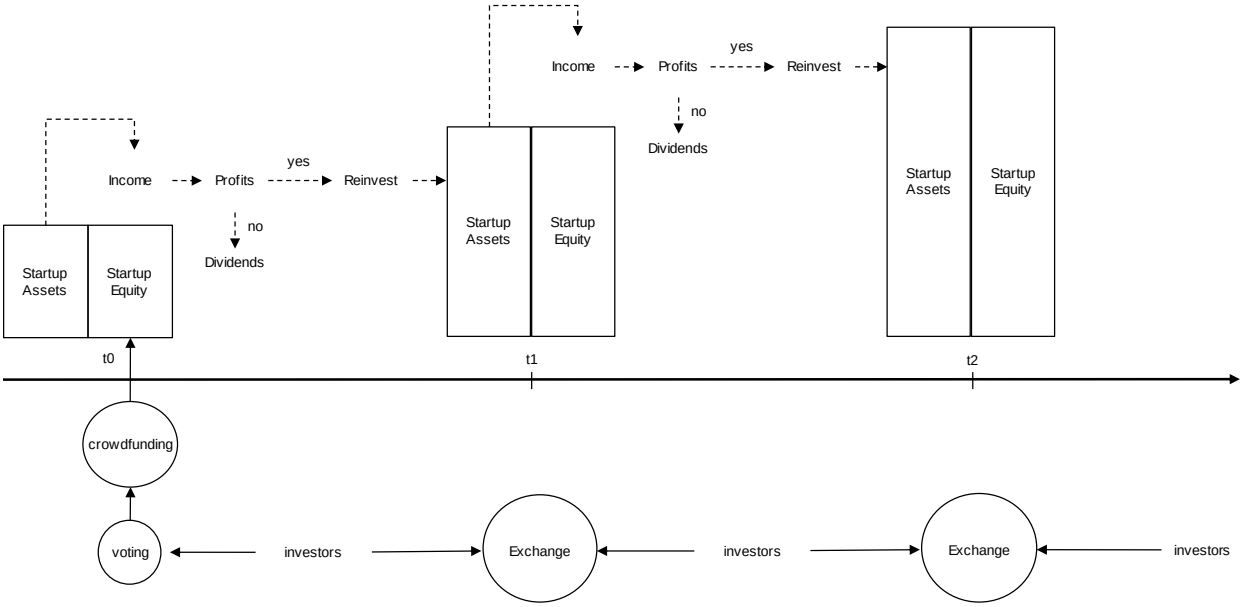
All rights reserved for mifuturofinanciero.com SpA

In the current low-liquidity markets, an early investor cannot sell their positions, so will want to receive some dividends, the quicker the better. But, for a small company to grow as rapidly as market opportunities allow, it is advisable to re-invest all the profits to fuel growth.

So, in the early stages of its growth, a small company does not have the capacity to “pay back” investors using the classical dividend yield policy. They should not pay dividends too soon after they start to earn profits. Later, after it fills its market niche and becomes more mature, it can pay dividends. Before that time, however, investors have, in the past, found that they are “locked in”.

A functioning secondary market for this kind of equity will mean that the startup’s investment and re-investment decisions can be independent of the need to “cash out” some of the early investors, or to pay dividends when the business has good opportunities to re-invest its earnings. For investors knowing that they can sell at any time, rather than thinking they need to hold the investment for several years before being able to harvest their gains, our platform can be a shift paradigm.

Dividend yield versus expected capital gain



To resolve those issues, we built a primary market that aligns economic incentives among all participants, attracts voters as a way of screening which startups will be allowed to issue, and a secondary market for these small blocks of equity, that allows investors' decisions to be independent of a holding period constraint.

The technical improvement that Godzillion accomplishes is that the startup Voting and Issuance processes (crowdvoting and crowdfunding campaign) and Investors' trades are executed using a Smart Contract Architecture on the Ethereum Blockchain.

This is a major breakthrough: there is no central server, but instead an extremely reliable voting reward, order confirmation and tracking system, verifiable publicly and updated in real time.

The result is a quantum improvement for Investors and Startups. Entrepreneurial proposals are now able to be voted in a way that has the correct economic incentives, and are then able to raise money with very low administrative and transaction costs. And the entire step-by-step process will be implemented with very high security in the settlement procedure, so investors will have a verified record of all steps of the capital raising process.

Using Godzillion, the Investors who want to cash out at any time, are able to post an ASK order for their blocks and, when another investor pays that ASK price, can then harvest their gains.

With the Godzillion DApp, Startups can be voted, issue tokens and achieve greater liquidity for their early Investors, on the Ethereum Blockchain, worldwide.

All rights reserved for mifuturofinanciero.com SpA

3. Rewards on Voting and no fees on Value.

Financial Intermediaries such as Investment Banks and Brokerage Firms charge their clients, in most cases, upfront fees, then fees as a percentage of the capital raised or traded, and a percentage of the assets under management. Their business model is to act as coordinators of the end-to-end process of investments and financing business, implementing trades and settlement between different parties around the world and different currencies, and offering to individuals different private ledgers to keep the records of investors' assets held in custody.

In the conventional crowdfunding business model, the processes are to screen Startups' proposals, then post the acceptable ones as auctions on their websites. Crowdfunding sites attract investors who open accounts on the sites, and then bid in the auctions posted on the sites.

When the auction reaches its goal, the crowdfunding site, moves the money the investors committed from the investors' accounts to the crowdfunding sites' bank account, issue proof of purchase to investors, record all the information in their servers, and transfer the money from their bank account to the issuer's bank account. The money transfer records are all stored on the bank´s server, and the issuing records are on the crowdfunding company's server.

In this traditional approach, there is often no procedure by which a secondary market trade involving the newly issued securities can be recorded and can be transferred in a fast and secure way. To do a trade, a seller and a buyer have to find each other, and then the records of ownership and payment on different servers have to be updated. This multi-step process can take more than 5 days to be settled and the associated costs can discourage issuance and trading.

The financial intermediaries that operate according to the classic crowdfunding business model receive income by charging upfront fees and fees on capital raised, as they trade securities issuance for cash. The total fee can be, in some cases, more than 10% of the capital involved. In this classic procedure. the fee is needed to cover the intermediary's operational costs; but viewing what the classic intermediaries do in a more prosaic way, is they screen proposals, and then when the buyer and seller both have accounts with the same intermediary, they transfer records from one ledger to another keeping the settlement process in their purview.

With the Godzillion DApp, the steps of pricing issuing and trading change. Investors have economic incentives to screen the Startup proposals and the fees are not computed as a percentage of the capital or value involved. Instead, the costs are driven by the computational steps required to execute the issuance or the trade in the Ethereum Blockchain.

			Centralized	Decentralized
Process	Voting / Screening	Issuer	upfront fee to the crowdfunding company	reward to voters screening
		Investor	--	financial return per economic vote
	Issuing	Issuer	% of the raised value to the crowdfunding company	computational steps
		Investor	% of the raised value to the crowdfunding company	computational steps
	Trading	Buyer	--	computational steps
		Seller	--	computational steps

In other words, fees are equivalent to the energy and computing power required to execute an order (subscribe or trade), not as a percentage of the value involved in the transaction. The cost (fee) is driven by the data computation (measured in bytes), not from the value that the data represents.

The change in the paradigm of how Startups are screened and how value is traded and recorded, implies a substantial reduction in operational and implementation costs, increases in market access to information, allows trading transparency and improves security of the issuance process.

All rights reserved for mifuturofinanciero.com SpA

4. Godzillion, a Smart Contract System on the Ethereum Blockchain

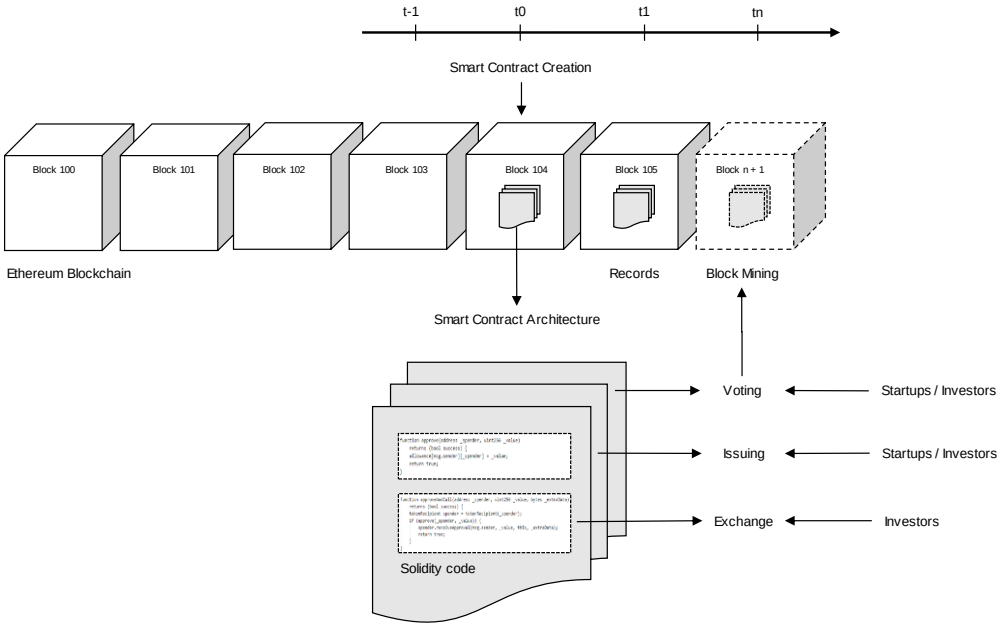
Ethereum Protocol operates as a decentralized virtual machine where Smart Contracts can be deployed in order to execute pre-defined lines of code that do specified tasks. The crypto currency that operates as cash in the Ethereum Blockchain is Ether (ETH) and the basic confirmation and settlement process is to transfer ETH from one account to another account, or to a Smart Contract, which will distribute the ETH according to instructions in the Smart Contract. In the Ethereum protocol, there is a defined and known cost structure for each kind of transaction that can be implemented using Smart Contracts and settled on the Ethereum Blockchain. Godzillion is a decentralized application (DApp) that is built as a Smart Contract on the Ethereum Virtual Machine. So, it is coded to execute clients' requests, while keeping the records in the Ethereum Blockchain. For instance, those requests can be Transfers, Voting, Crowdfunding Campaigns and Trades.

The confirmation service providers (called miners) who validate the transactions (blocks) and keep an up-to-date copy of the entire Blockchain earn rewards. The miners earn rewards, both from solving a difficult math problem (Proof of Work) to validate an entire block of transactions (5.0 ETH per winning block), and all the gas expended within the block, that is, all the gas from the contracts that were run within the block submitted by the winning miner. To achieve this goal, the miners run computers (hardware) and nodes (Ethereum Virtual Machine) around the world competing to confirm transactions and get ETH in reward (today the confirmation process is executed as a proof of work; in the near future, the process probably will be as a proof of stake).

The competition among miners is lively, and the rewards give economic incentives to ensure that there will be many service providers, so the result is a decentralized Virtual Global Confirmation Machine that is a key attribute of the Ethereum Virtual Machine. The number of active nodes has value, because the capacity to confirm and store data, is now full decentralized. The Ethereum Blockchain does not reside in a single server, instead, it is a single public ledger, stored on different nodes in every time zone, where miners compete to confirm transactions and create a new State of the Ledger (incorruptible book keeping system).

All Godzillion transactions are executed, recorded and stored on the Ethereum Blockchain by the Smart Contracts Architecture, so the executions requests need "gas" in order to be processed and data be recorded into the Blockchain. The service providers (called "miners"), that confirm transactions ordered by Smart Contracts transactions and update the Blockchain, charge for performing that service, and they are paid with "gas".

So, the "gas" that is needed to operate a Smart Contract depends on how complex the transaction is, or how many instructions the Smart Contract includes, or how much space on the Blockchain is required. The more complexity in the request, the more gas needed. The "gas" fee is not calculated as a percentage of the value involved. Instead, is aligned to the execution type request to be implemented on the Ethereum Blockchain.



So, in order to create a functional global micro finance market for Startups, voted, executed and recorded in a global ledger, Godzillion's design includes a Token that operates with the Smart Contract Architecture. This Token is named GODZ and allow Godzillion users to vote on Startups and earn rewards for voting, and the Token enables the Startup market with ETH and it improve performance in trade execution using GODZ to swap Startups Tokens. Because Godzillion is a fully decentralized application, all the records and settlement details are executed in accord with Ethereum practices, and are added to the Ethereum blockchain.

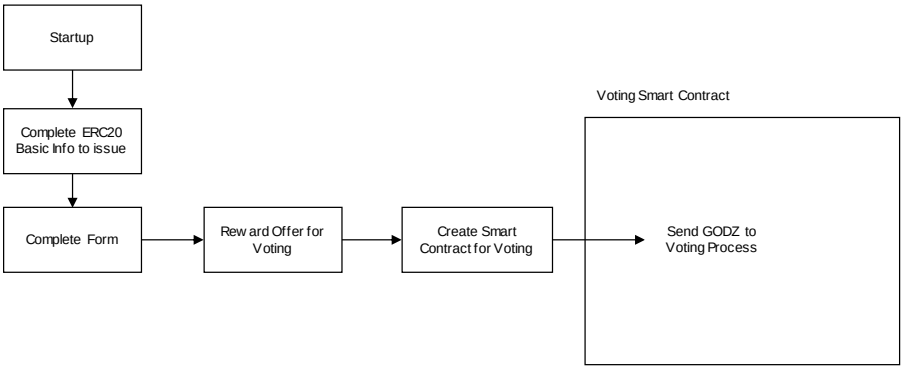
All rights reserved for mifuturofinanciero.com SpA

5. GODZ, the Token, what for?

GODZ allows users to vote whether a Startup proposal should be allowed to seek funding on Godzillion; it also allows users to buy the Alternative Assets Tokens that will be issued on Godzillion; and it facilitates trading those assets. The Smart Contract Architecture use GODZ to enable voting, issuance, distribution and control of the assets in a formalized and standardized manner. It minimizes costs and speeds up execution, while being fully recorded in the Ethereum Blockchain with 100% trace-ability.

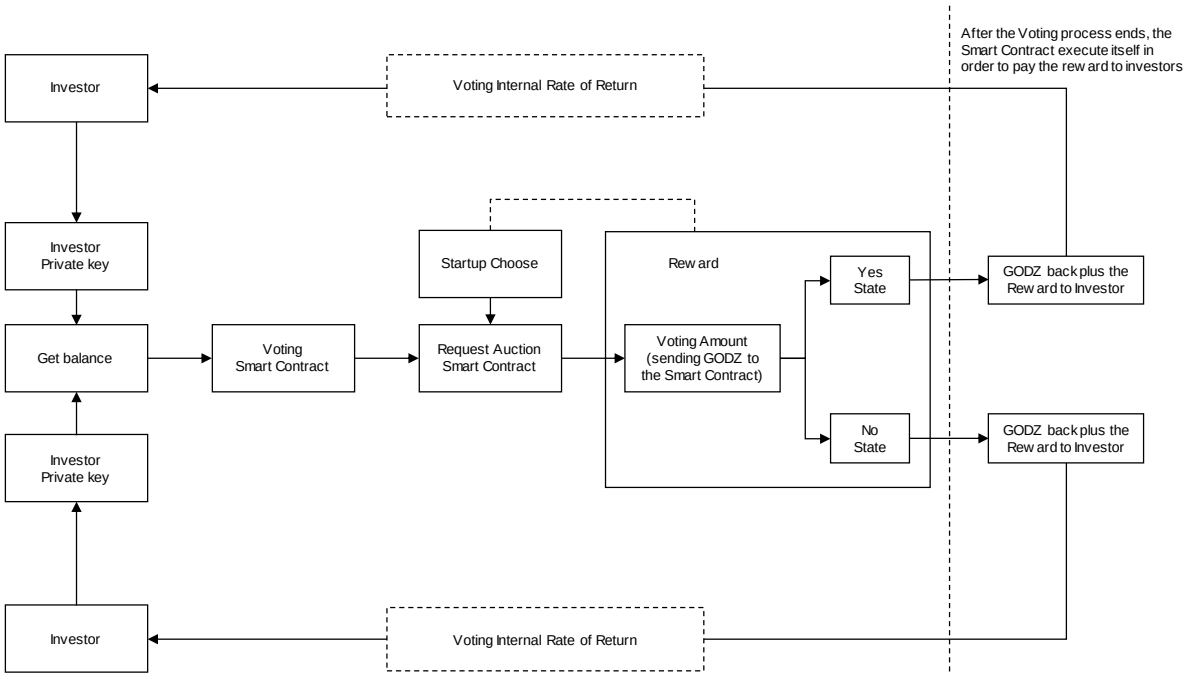
5.1 Voting / Screening:

Startups need to be screened (voted Yes by the crowd) to gain access to an auction process and issue their tokens. In order to be published and attract voters, a Startup creates a Voting Smart Contract and provides it with some amount of GODZ as an economic incentive to the voters. This amount of GODZ will be available to the winning voters (yes or no) as a reward after the auction period ends, whatever the result is.



Using the godzillion dapp, Startup CEOs complete a Form, define the reward and the amount in GODZ to be raised as the percentage of the company´s total equity (percentage that will be offered to investors via crowdfunding). After those parameters of the issuance have been set, the Voting process starts and investors can vote for the startup to be allowed to post an auction, or vote to reject the proposal. Investors vote by sending GODZ to the Voting Smart Contract.

The voting procedure is that a GODZ token holder (an investor) studies the proposal posted by the Startup, then sends GODZ to the Voting Smart Contract. Token holders can set the amount of GODZ they send to the Voting Smart Contract, and designate whether the GODZ they send are in favor of allowing the proposal to go forward, (a YES vote) or to reject it (a No vote). Then, when the voting process has ended, the Voting Smart Contract will transfer the GODZ back to the token holders (investors). The ones who voted for the winning side get their GODZ back plus the 95% of the GODZ that the Startup put into the Voting Smart Contract, as an upfront fee for being screened; the token holders (investors on the losing side) get their GODZ back plus the 5% of the reward.

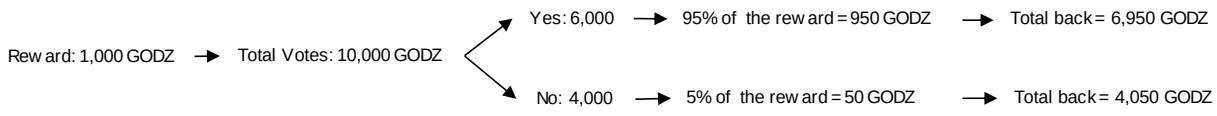


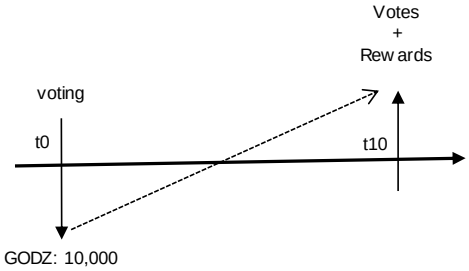
All rights reserved for mifuturofinanciero.com SpA

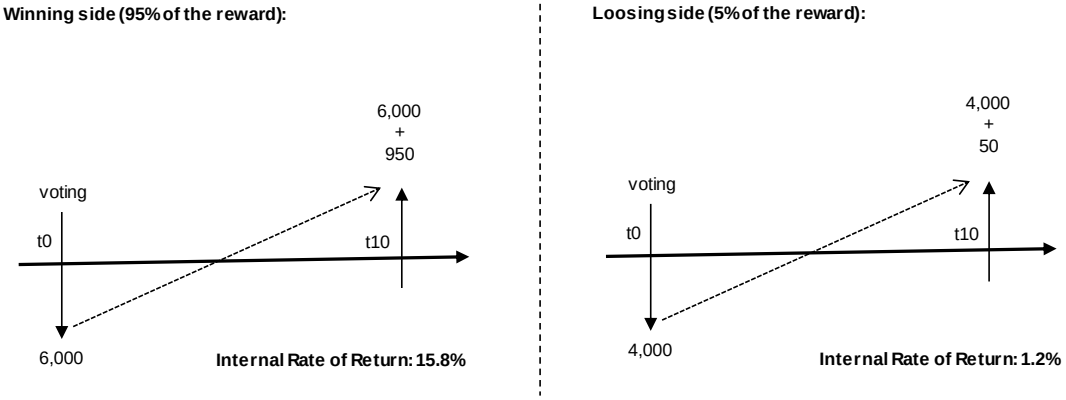
So, Investors have an incentive to vote, either to approve the Startup to raise capital and issue tokens through godzillion, by voting yes; or if they think the proposal should be rejected, by voting no. The voting period is 10 days; each Startup has 10 days when the voters give their opinions. Voters will be compensated for voting, even if they vote for the losing side.

For example, if a Startup offers a reward to voters of 1,000 GODZ in order to be screened, and in a period of 10 days the voters send a total of 10,000 GODZ, of which 6,000 GODZ went to the Yes vote, the winners get their 6,000 GODZ back plus the 95% of the reward, 950 GODZ.

In financial terms, winning voters get a 15.8% return on investment voting on the Startup. Losers get their 4,000 GODZ plus 50 GODZ, receiving a 1,2% return. This return is computed for the voting period of 10 days, which is the period when the Startup's proposal is being voted on by investors. So an investor who buys GODZ and uses them to vote on proposals can earn a high annual rate of return. The return will be very high for an investor who can predict which proposals are going to be approved and which ones are going to be rejected.







With this logic, Godzillion its able to create economic incentives to attract investors to participate in the Startup decentralized screening process.

The fee that investors can earn by participating in voting, in the classic institutional framework, used to be paid to an Investment Bank or a Crowdfunding Company. Now, the Startup offers a fee to the market as a reward for voting on its proposal, and to be approved, or disapproved, to issue their tokens.

Because the internal rate of return of participants in the voting process depends on the Startup reward and the amount as votes on each side of voting, we expect participants will compete to reach an economic equilibrium in voting pricing.

All rights reserved for mifuturofinanciero.com SpA

In order to illustrate in detail how Godzillion implements the crowdvoting processes, we present below part of the Smart Contract code. We show the function step by step. It is called GodzStartupsVoting.

This first part of the dapp sets up the structure of the voting process for a Startup. In the step called struct declare StartupVotingStruct we gave the structure to store the tokens in the Voting Smart Contract. The index allows holding only Smart Contracts from Startups that are in the voting process.

```
function GodzStartupsVoting(address _godzContract)
{
    owner = msg.sender;
    godzContract = _godzContract;
}

struct StartupVotingStruct
{
    address startupContract;
    uint256 initialVotingDate;
    uint index;
}
```

We request a Startup Smart Contract from the Voting Smart Contract, and we validate that the startup is registered into the Voting Smart Contract administrator, for the voting process. We also made a validation about the number of Startups that are in the voting process.

```
function getStartupAtIndex(uint index)
public
constant
returns(address startupContract) {
    return startupVotingIndex[index];
}

function getStartupCount()
public
constant
returns(uint count) {
    return startupVotingIndex.length;
}
```

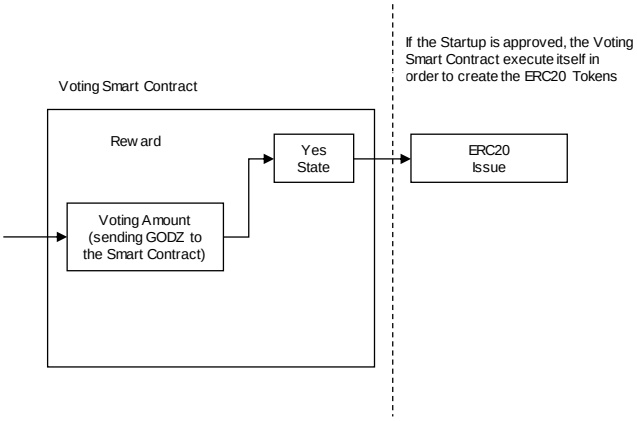
This is part of the function is the one that pay the reward to the investors. In this case, if the voting was No, we made a calculation process in order to transfer the reward, depending if the No was the winning side of the voting or if was the losing side. In case that is the winning side, investors are paid with the 95% of the reward that the startup company put as incentive for the voting process. In case that the No was the losing side, investors are paid with the 5% of the reward. After the calculation process, the smart contract transfers the reward to the investors.

```
if (voteNo[startupContract][voter]>0)
{
    if (godzVotingTotalYes[startupContract]<godzVotingTotalNo[startupContract])
    {
        premioGodz = (GodzStartupInformation(startupContract).reward() * 95)/100;
    }
    else
    {
        premioGodz = (GodzStartupInformation(startupContract).reward() * 5)/100;
    }

    transferir = ((percent(voteNo[startupContract][voter],godzVotingTotalNo[startupContract],3) * premioGodz)*1000000000000000000)/1000;

    return (voteNo[startupContract][voter], transferir, premioGodz);
}
```

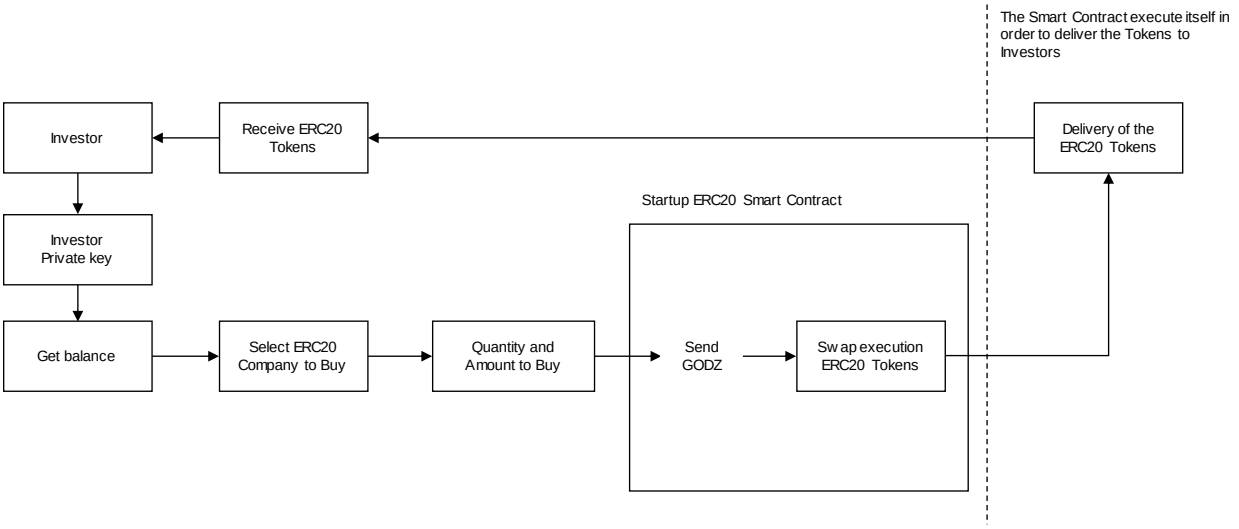
All rights reserved for mifuturofinanciero.com SpA

5.2 Auction / crowdfunding campaign:

After the screening process, if the Startup is approved, it Auction process starts. Because auction process is also executed using GODZ, it is implemented and recorded using Smart Contract invocations. This process is transparent and can be audited by anyone using a blockchain explorer.



In this category, there will appear all of the companies that are in auction process. The investors then select which auctions to they want to bid in. This process involves two main outcomes: the investors receive their tokens, and the Startup receive the capital that it applied for (GODZ). Tokens are delivered to the Investors' Ethereum Wallets and the GODZ which were raised are delivered to the Startup's Ethereum Wallet.



To summarize, once the investor selects the quantity of Startup tokens that he wants to bid, and validates the GODZ balance necessary to buy those tokens, and that moment, went the investor clicks on Buy Tokens, he will send the GODZ to the Startup Smart Contract and the Smart Contract will automatically send the tokens to the Investor's wallet and the GODZ to Startup's wallet (Swap).

This auction process will end in the moment that the total quantity of tokens had been sold, no matter if takes 1 day or 60 days. At that point, the startup has its GODZ convertible into ETH and Investors who want to sell their equity holdings of the Startup can post an ASK order to the godzillion decentralized market.

In order to implement those processes, we declare the variables that we request as basic info for the ERC20 contract. That is in the form that Startups complete as a prerequisite for commencing the voting process.

This information that we have collected, and which the Startups have provided, is the information that we use to issue the ERC20 contract for the voting process. Once that declaration, and the input variables for the function that collects the information from the form are complete, we create two linkage relationships: (i) between the wallet of the startup that is requesting the voting, and the ERC20 contract information; and (ii) between the wallet and the information about the valuation information of the Startup.

All rights reserved for mifuturofinanciero.com SpA

This last information is part of a Part A section that we divide in order to save all the information of the Startup in a struct declaration.

```
function insertRelationStartup(
        string    _startupName,
        string    _startupSymbol,
        uint256   _startupSupply,
        uint256   _startupAmount,
        uint256   _startupReward,
        uint256   _startupPercentage,
        uint256   _startupValuation,
        uint256   _startupTokenPrice
    )
{
    address _GodzStartupBasicInformation = new GodzStartupBasicInformation(
            _startupName,
            _startupSymbol,
            _startupSupply,
            _startupAmount,
            _startupReward
        );

    RelationWalletGodzStartupBasicInformation[msg.sender] = _GodzStartupBasicInformation;

    address _GodzStartupExtendedInformation = new GodzStartupExtendedInformation(
            _startupPercentage,
            _startupValuation,
            _startupTokenPrice,
            _GodzStartupBasicInformation
        );

    RelationWalletGodzStartupExtendedInformation[msg.sender] = _GodzStartupExtendedInformation;
}
```

In the voting process section, when an investor need to review the information of the Startup, a request to the smart contract is made that holds the information of the Startup. With the function getStartupInformationPartA we request the information of a certain Startup that the investor request once that select the company in the voting process. With the return function, the information is deployed in the DApp so the investor can analyze it.

```
function getStartupInformationPartA(address _startupContract)
public
constant
returns (
        uint256   _startupPercentage,
        uint256   _startupValuation,
        uint256   _startupTokenPrice,
        string    _valueOfwebsite,
        string    _valueOfnameCEO,
        string    _valueOflinkedinCEO,
        string    _valueOfcountry
    )
{
    return (
        StartupIssuingInfoStructs[_startupContract].startupPercentage,
        StartupIssuingInfoStructs[_startupContract].startupValuation,
        StartupIssuingInfoStructs[_startupContract].startupTokenPrice,
        valueOfwebsite[_startupContract],
        valueOfnameCEO[_startupContract],
        valueOflinkedinCEO[_startupContract],
        valueOfcountry[_startupContract]
        );
}
```

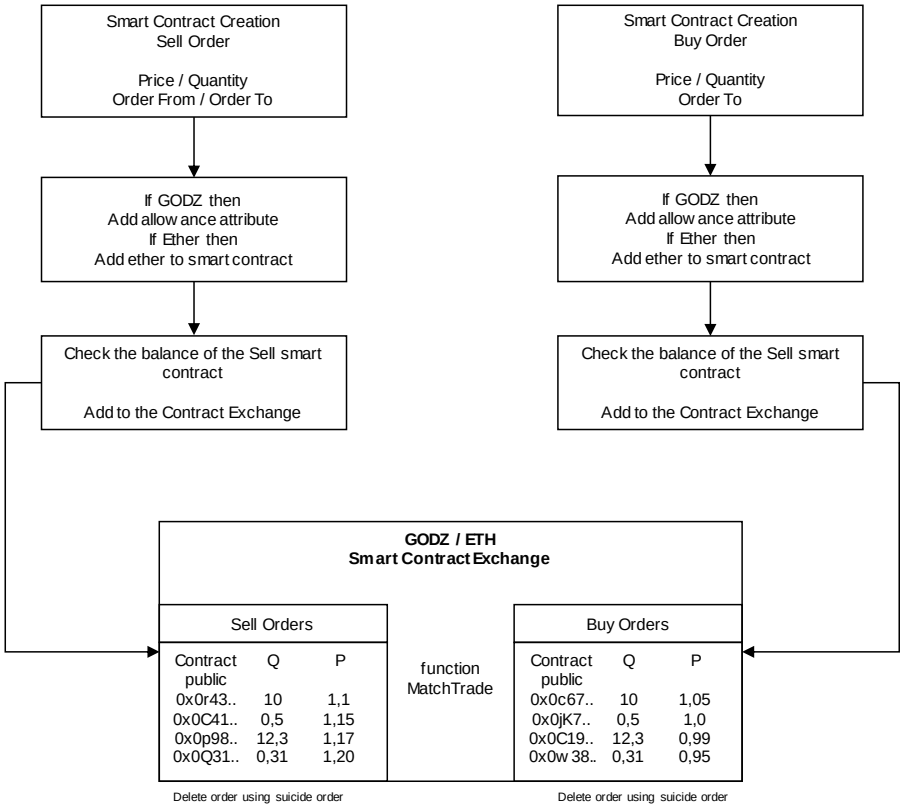
5.3 Exchange:

GODZ can be exchangeable at Godzillion DApp with ETH, also exchangeable with the Tokens of a specific Startup (like AAIF), and can also be exchanged for Ethereum issued Tokens (like GNO and REP). In Godzillion there are three decentralized markets: GODZ / ETH, Startups Tokens / GODZ; and Ethereum Tokens / GODZ.

All rights reserved for mifuturofinanciero.com SpA

5.3.1 GODZ / ETH Market

Investors can buy GODZ using ETH at the GODZ Decentralized Exchange. Investors can trade ETH against GODZ. In this exchange, with the balance of GODZ available in your wallet and the balance of ETH, Investors can post orders to buy or sell GODZ.



As we describe in the next code, we declare a struct architecture of Smart Contracts to create the exchange of GODZ for ETH. The variables that are used in different Smart Contracts that compose the Godzillion exchange are declared in this part of the Smart Contract. Next, whit the mapping we arrange the order structure for the Smart Contract for a Buy or Sell order.

```
struct MarketOrderStruct
{
    address tokenFrom;
    address tokenTo;
    address userAddress;
    uint quantity;
    uint price;
    uint index;
}

mapping(address => MarketOrderStruct) private MarketOrderStructs;

address[] private marketOrderIndex;
```

This if is part of the executeTrade function that we create for the exchange. If there is a trade and validate that the quantity of a trade is bigger than 0, we insert the trade into the trade history Smart Contract, that save all the trade information executed into the exchange.

```
if (trade(sellOrder, buyOrder, amount))
{
    if (quantity>0)
    {
        GodzExchangeTradeHistory(tradeHistory).insertTrade(tradeAddress, fecha, tipo, precioMenor, quantity, amount);
    }
}
```

All rights reserved for mifuturofinanciero.com SpA

If the quantity of the sell order its exactly the same of the buy order, then this function is executed. Here we transfer the quantity of tokens from the seller to the buyer, given that there exists a match between the two orders. After we transfer the tokens, we make an update to the order book, updating the order book to include the trade that was executed.

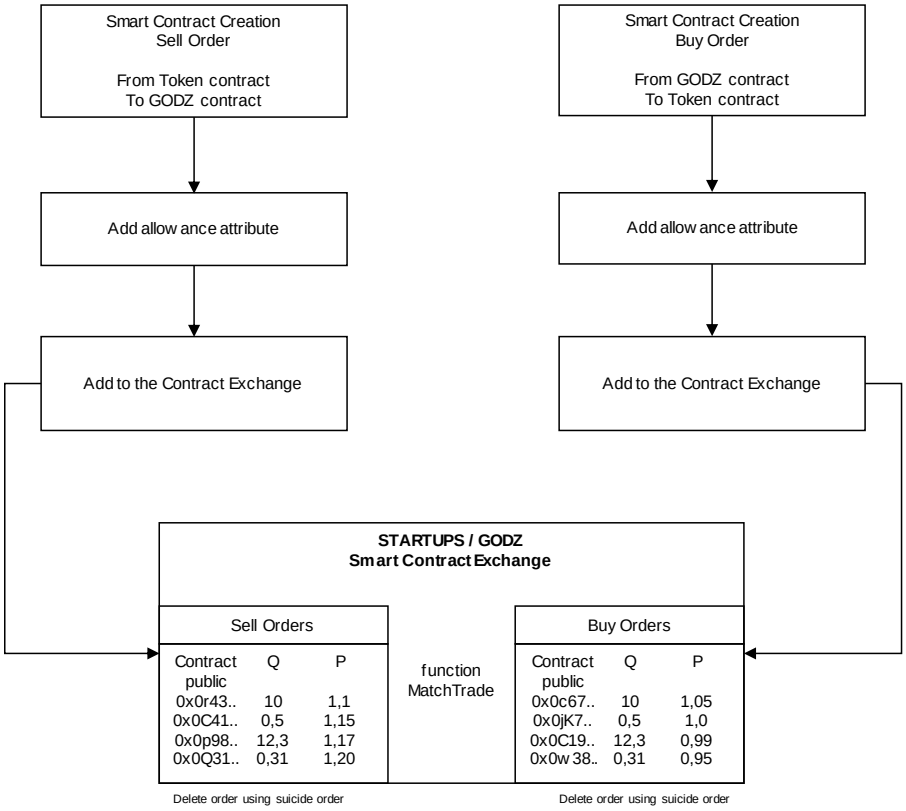
```
if (MarketOrderStructs[sellOrder].quantity==MarketOrderStructs[buyOrder].quantity)
{
    Token(MarketOrderStructs[sellOrder].tokenFrom).transferFrom(
        MarketOrderStructs[sellOrder].userAddress,
        MarketOrderStructs[buyOrder].userAddress,
        MarketOrderStructs[sellOrder].quantity
        );

    marketBuyOrderGodz(buyOrder).transfer(MarketOrderStructs[sellOrder].userAddress,amount);

    marketBuyOrderGodz(buyOrder).changeQuantity(0);
    marketSellBuyOrderGodzToken(sellOrder).changeQuantity(0);

    updateOrderQuantity(buyOrder,0);
    updateOrderQuantity(sellOrder,0);

    return true;
}
```

5.3.2 Startups Tokens / GODZ Market

When the auction process has been closed, the tokens issued by the Startup will immediately be available to be trade in Godzillion's Startup Market for Startup Tokens. Using GODZ, investors can buy or sell tokens of the Startup they have participated in, or tokens of other Startups that are already listed on the decentralized exchange.

By selecting the token that Investor want to trade, he will have access the public address of the ERC20, and also to a button where the information of this smart contract can be review on etherscan.io.

With the balance of GODZ available in the Investor wallet and the balance of tokens of the selected company, Investors can post orders to buy or sell the Startup Token.



All rights reserved for mifuturofinanciero.com SpA

In this case, the function that we use to validate if the address of the Smart Contract is an buy or sell order from the exchange, is using the fix index. If the fix index is 0, there is not completed order yet, as we describe in the next code:

```
function isOrder(address userMarketOrderContract)
    public
    constant
    returns(bool isIndeed) {
        if (marketOrderIndex.length == 0) return false;
    return (marketOrderIndex[MarketOrderStructs[userMarketOrderContract].index] == userMarketOrderContract);
}
```

For the function created to insert an Buy or Sell order into the exchange, first we validate if the order exists in our exchange. If it exists then throw.

Then we store the public address of the wallet that issues the order, along with the quantity, price, token from, token to, and the index of the order in the exchange. With this process, the order is stored in the exchange, waiting for a match to trade.

```
function insertOrder(
    address userMarketOrderContract,
    address userAddress,
    uint quantity,
    uint price,
    address tokenFrom,
    address tokenTo)
    public
    returns(uint index) {
    if (isOrder(userMarketOrderContract)) throw;
    MarketOrderStructs[userMarketOrderContract].userAddress = userAddress;
    MarketOrderStructs[userMarketOrderContract].quantity = quantity;
    MarketOrderStructs[userMarketOrderContract].price = price;
    MarketOrderStructs[userMarketOrderContract].tokenFrom = tokenFrom;
    MarketOrderStructs[userMarketOrderContract].tokenTo = tokenTo;
    MarketOrderStructs[userMarketOrderContract].index = marketOrderIndex.push(userMarketOrderContract) - 1;
    return marketOrderIndex.length - 1;
}
```

If a quantity of a buy or sell order needs to be updated, for example, because it was a trade for this order, but the seller or buyer doesn't take all the quantity of the order. For example, if somebody sells 20 tokens, but someone buys from him only 10, the sell order now is updated for 10 tokens remaining to sell.

Here part of the function,

```
function updateOrderQuantity(address userMarketOrderContract, uint quantity)
    public
    returns(bool success)    {
    if(!isOrder(userMarketOrderContract)) throw;
    MarketOrderStructs[userMarketOrderContract].quantity = quantity;
    return true;
}
```

5.3.3 Ethereum Tokens / GODZ Market

If an Investor don’t have either ETH or GODZ in their wallet, but they have other types of tokens that had been issued in the Ethereum Blockchain, they can also trade those tokens for GODZ.

In order to do that, in this section Investors must add a Token with the public address of the ERC20 (token issued in Ethereum Blockchain), then click on the button Validate ERC20. The dapp will search the information of that ERC20 in the Blockchain, and will show the Name associated with that ERC20, the Total Supply of tokens, the Symbol and the Decimals.

All rights reserved for mifuturofinanciero.com SpA

If the information that appears is correct, then, Investors can click the button Add Token, and the dapp will add this token to the Smart Contract Exchange.

So, once the process has ended, Investors are able to trade these tokens with GODZ. When this exchange has at least one token to trade, Investors can select the token that he wants to trade. After the investor selects, the dapp shows the public address of the ERC20, and the balance of tokens that the Investor has in his wallet.

Also, it will show the balance of GODZ that the Investors has. With the balance of GODZ available in the Investor wallet and the balance of added tokens, Investors can post orders to buy or sell the Tokens.

As part of the exchange, the struct declaration is used because different Smart Contract interact in order to create this exchange. In this case, the structure for the Token Exchange, use the mapping to order the tokens that can trade in this exchange.

```
struct MarketTokenStruct
{
    address exchange;
    address trades;
    uint index;
}

mapping(address => MarketTokenStruct) private MarketTokenStructs;

address[] private marketTokenIndex;
```

We use getTokens for obtaining the exchange Smart Contract and history trades for a Token that is listed in our exchange. This is used to obtain information of buy or sell Smart Contracts from the exchange or the history of trades that were executed.

```
function getToken(address erc20)
    public
    constant
    returns(address exchange, address trades, uint index) {
    return (
        MarketTokenStructs[erc20].exchange,
        MarketTokenStructs[erc20].trades
        MarketTokenStructs[erc20].index);
}
```

The index order is for order the information request inside the exchange. In the first function, the market order Smart Contract its called to obtain the index of the token inside that contract. In the second function, a count of the amount of tokens of the exchange for a token selected it's made.

```
function getTokenAtIndex(uint index)
    public
    constant
    returns(address userMarketOrderContract) {
    return marketTokenIndex[index];
}

function getTokenCount()
    public
    constant
    returns(uint count) {
    return marketTokenIndex.length;
}
```

All rights reserved for mifuturofinanciero.com SpA

5.4 Transfers and History:

With the Balance & Transfer service, users can review the GODZ, tokens and ETH balance of their wallet. Also, they can make transfers of GODZ, ETH and others Ethereum tokens, to other wallets.

In order to execute a transfer of ERC20 tokens to other wallets, the value of the transfer and the contractfrom are used to allow the wallet to transfer the quantity of tokens to another wallet.

The information is sent to implement the transfer according with the decimals that the token was issued, because there are different types of token listed in our exchange.

Also, users can review all the information related to the trades made on Godzillion dapp with their private key. They can review the date, the type of transaction (if it was a buy or a sell transaction), the value associated with the trade and finally the token traded.

Here part of the code,

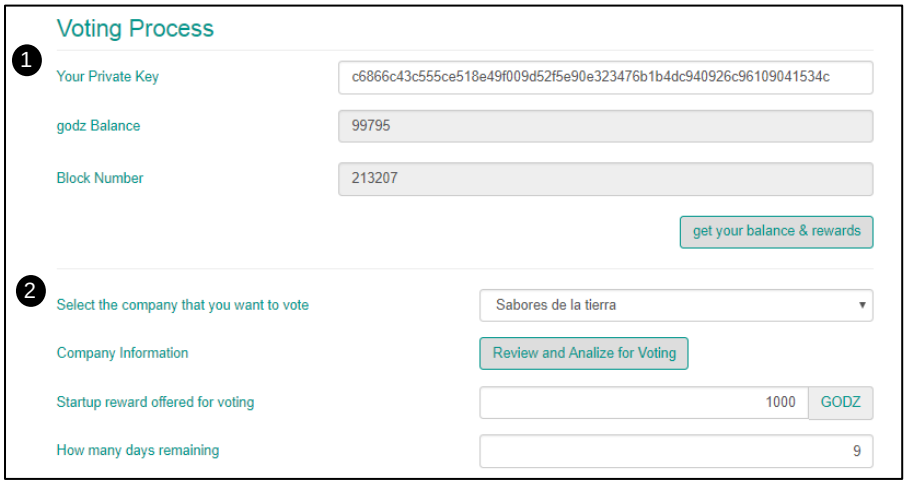
```
window.transERC20 = function (to, value, ContractTokenFrom) {
    var contractAddress = ContractTokenFrom;

    var solidityFunction = new SolidityFunction('', _.find(abiToken, { name: 'transfer' }), '');

    var contractPow = web3.eth.contract(abiToken).at(ContractTokenFrom);
    var powToken = Math.pow(10, contractPow.decimals.call().toNumber());

    var valueSend = (value * powToken);

    var payloadData = solidityFunction.toPayload([to, valueSend]).data;
    var nonce = web3.eth.getTransactionCount('0x'+address);
    var nonceHex = web3.toHex(nonce);
    var gasPrice = web3.eth.gasPrice;
    var gasPriceHex = web3.toHex(gasPrice);
    var gasLimitHex = web3.toHex(4700000);
    var privateKey = returnBuffer($('#pk').val());
    var rawTx = {
        nonce: nonceHex,
        gasPrice: gasPriceHex,
        gasLimit: gasLimitHex,
        to: contractAddress,
        from: '0x'+address,
        data: payloadData
    };
    var tx = new Tx(rawTx);
    tx.sign(privateKey);
    var serializedTx = tx.serialize();
    web3.eth.sendRawTransaction('0x' + serializedTx.toString('hex'),
        function (err, hash) {
            if (!err) {
                $('#trx').val(hash);
                waitForTransactionReceipt(hash);
            }
        });
}
```

All rights reserved for mifuturofinanciero.com SpA

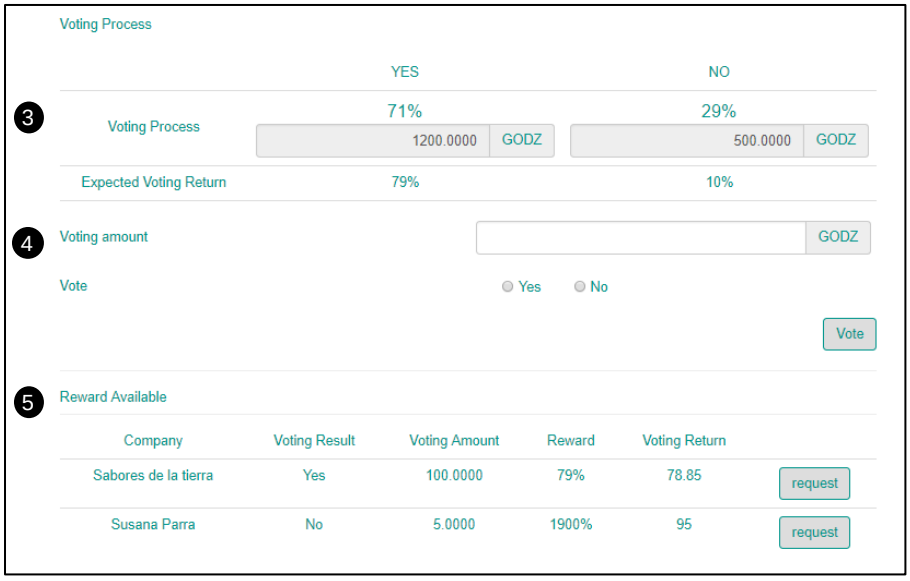
6. State-of-the art

6.1 Voting:

1) The investor needs to put the private key of his wallet here, in order to activate this section of the DApp. Here, the investor can review his GODZ balance and the last Block Number of Ethereum Blockchain.
2) The investor next can select the Startup that he wants to vote on, and with the Startup selected, there will appear the reward that the company has paid to be screened, and how many days remain before this voting process ends. The investor can click the “Review and Analyze for Voting” button to read all the information that the company completed in the form that it filled out in its application.



3) Here the investor can review how the voting process is going for the selected company, with percentage of the Yes or No voting, the amount of GODZ in each option and the Expected Voting Return.
4) The investor now can put the amount of GODZ that he wants to allocate in the voting process, and then select his option (Yes or No), and then click on the “Vote” button.
5) Finally, in this table the investor can review the rewards available, what the rewards have been in the past, and can request and review the information about the companies that ended their voting processes, and to see was the return to investors who participate in the voting process.

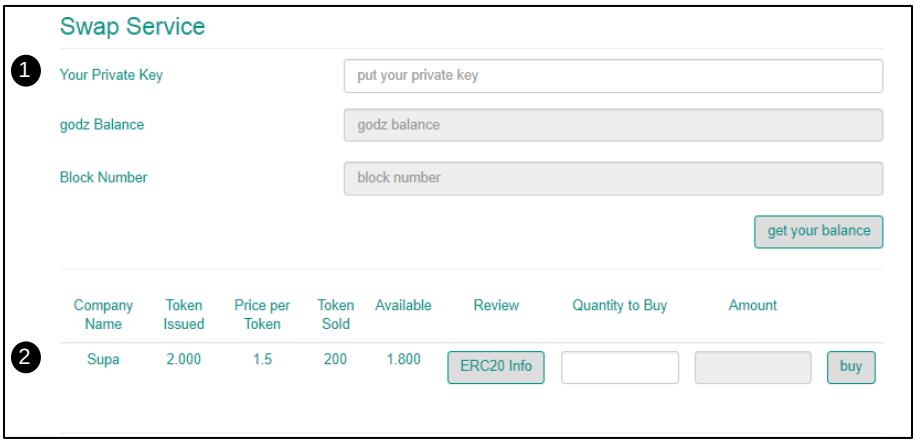


All rights reserved for mifuturofinanciero.com SpA

6.2 Swap and Issue:

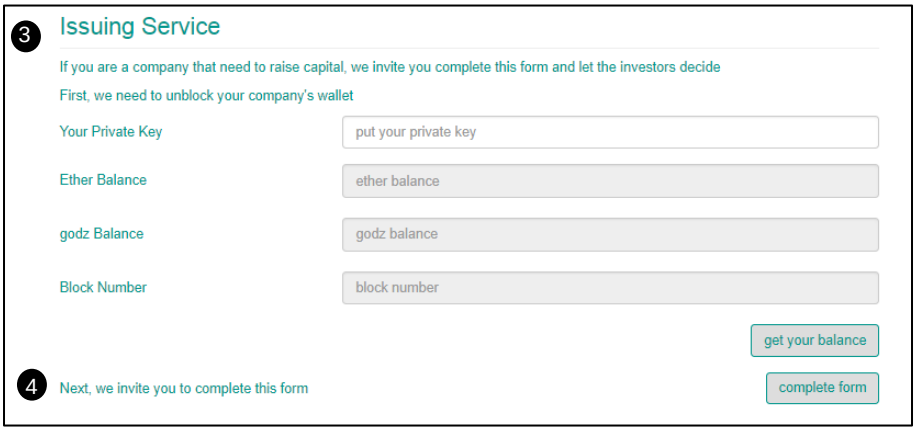
1) Investor needs to put the private key of his wallet here, in order to activate this section of the DApp. Here, the investor can review his GODZ balance and the last Block Number of Ethereum Blockchain.

2) In this table will appear all the startup companies that had positive voting, that means that investors vote Yes and with that, the ERC20 Token issued by the company will appear in this section. In this case we have Supa, that issued 2.000 tokens at 1,5 GODZ per Token. Has sold 200 and have 1.800 available to swap. If the investor wants to review the ERC20, he can click on the button. Then, the investor just need to put the amount of tokens that he wants to buy. The total amount will appear after that, and by clicking on "buy" he will add this token to his wallet.

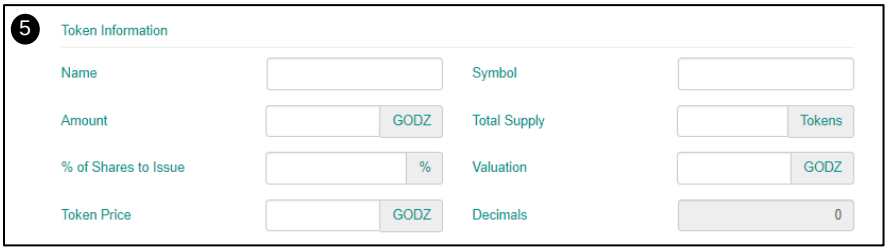


3) In order for a startup company to issue tokens through this DApp, it needs to first complete the form that we will present next. The company needs to have a Ethereum Wallet with balance of ETH and GODZ in order to proceed with the process.

4) Once we validate that the company's wallet has a balance of ETH and GODZ, the Startup can next complete the form with the information that we request.

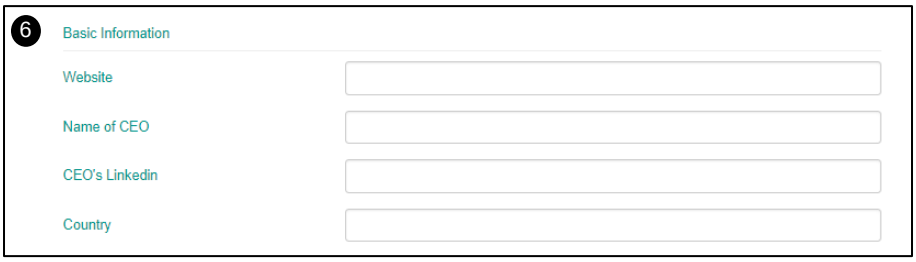


5) Here the company needs to complete the information related to the ERC20 Basic Info, and the information related to the amount to raise, percentage of the company to issue and the valuation associated with that percentage. By defect, the decimals are 0, because the token issue will be related to the shares of the company.

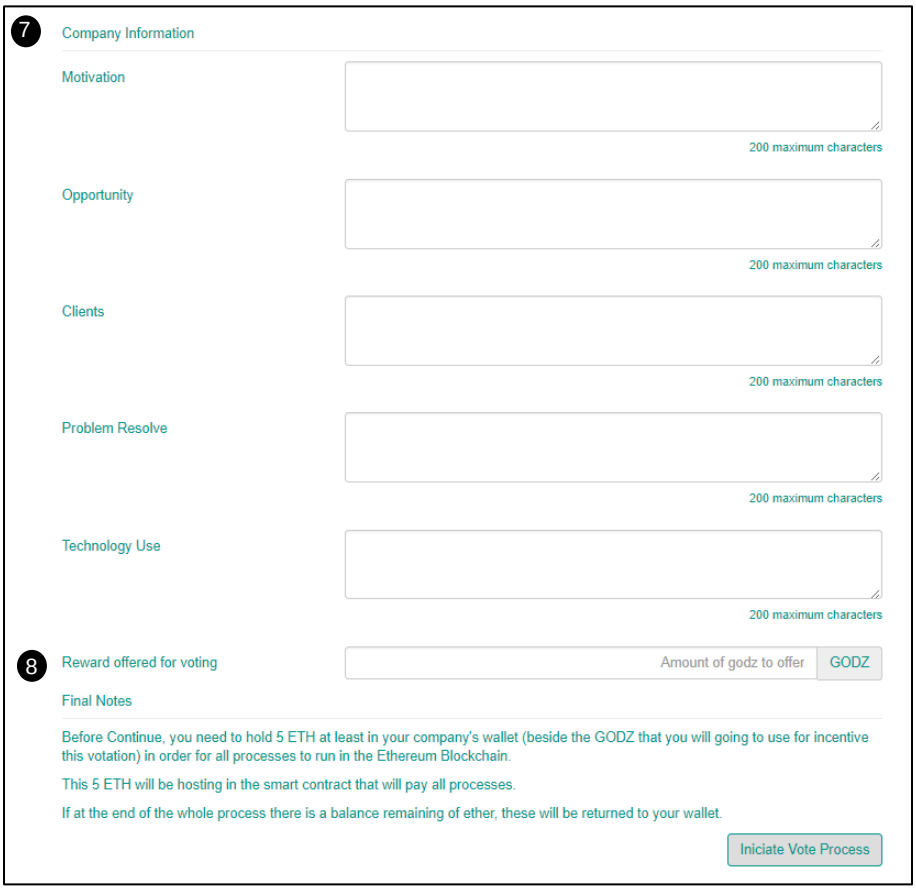


All rights reserved for mifuturofinanciero.com SpA

6) Next, the CEO will need to complete the Basic Info of the company, like the website, the name of the CEO, his Linkedin profile, and the Country.



7) The last set of information that the company needs to complete is related to its business.
8) Finally, with all the information completed, the company needs to put the amount of GODZ that will be offered as reward for the voting process to investors that will participate. After completing the application process, the company needs to click on “Initiate Vote Process”.



6.3 Exchange

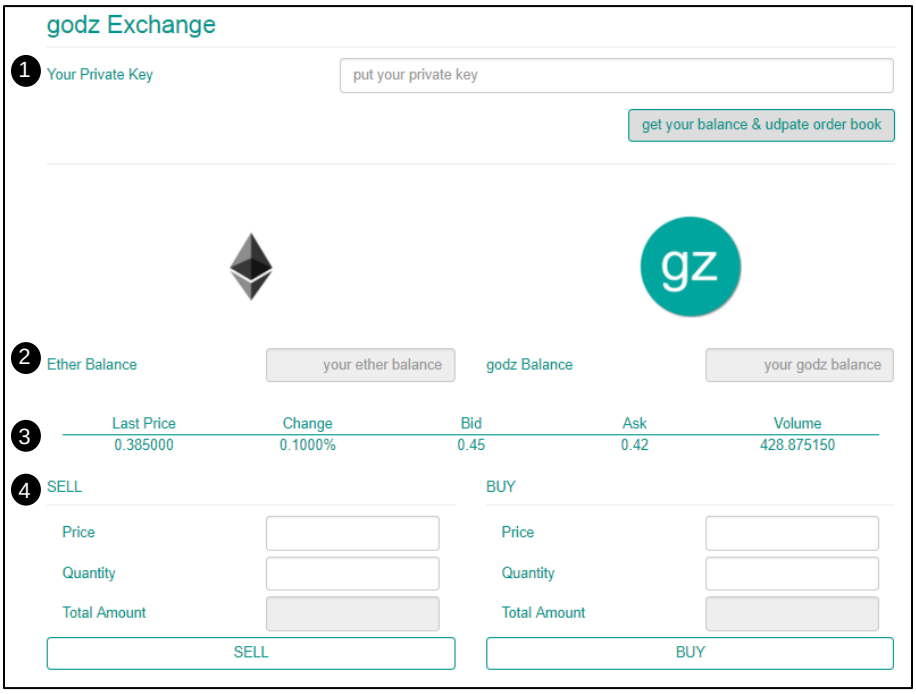
6.3.1 GODZ

1) The Investor needs to put the private key of his wallet here, in order to activate this section of the DApp
2) Once that investor put his private key, these fields will show his balance of ETH and GODZ. If it’s his first time on Godzillion, the balance of GODZ will be 0.

All rights reserved for mifuturofinanciero.com SpA

3) Here the investor can review the information of the last trade of ETH for GODZ executed in this exchange.

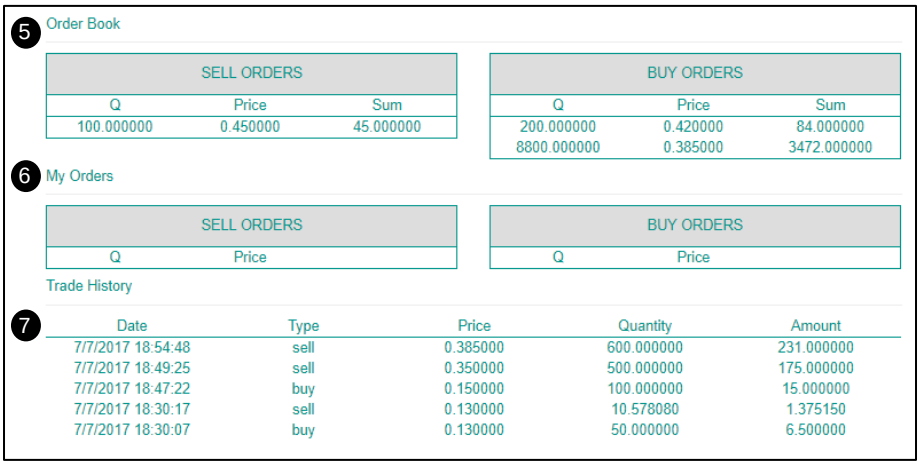
4) If the investor wants to buy GODZ, he needs to review the information of the last trade, or view the order book, to put in his buy order. The investor needs to put the price at which he wants to buy, the quantity, and total amount of ETH that he needs to include to place the buy order; then he can click the BUY button, and if the validation is ok, the order will be placed. In case if he wants to sell GODZ, and he has GODZ in his wallet, he needs to do the same steps, but click un the SELL order button.



5) Here the investor can review the Order Book of this exchange, to analyze the Bid and Ask, to place his Buy or Sell order.

6) In this section, the investor can review the orders that he has placed in the exchange, according to whether he wants to Sell GODZ, or if he wants to Buy GODZ. He can delete those orders if he wants, because they have not executed.

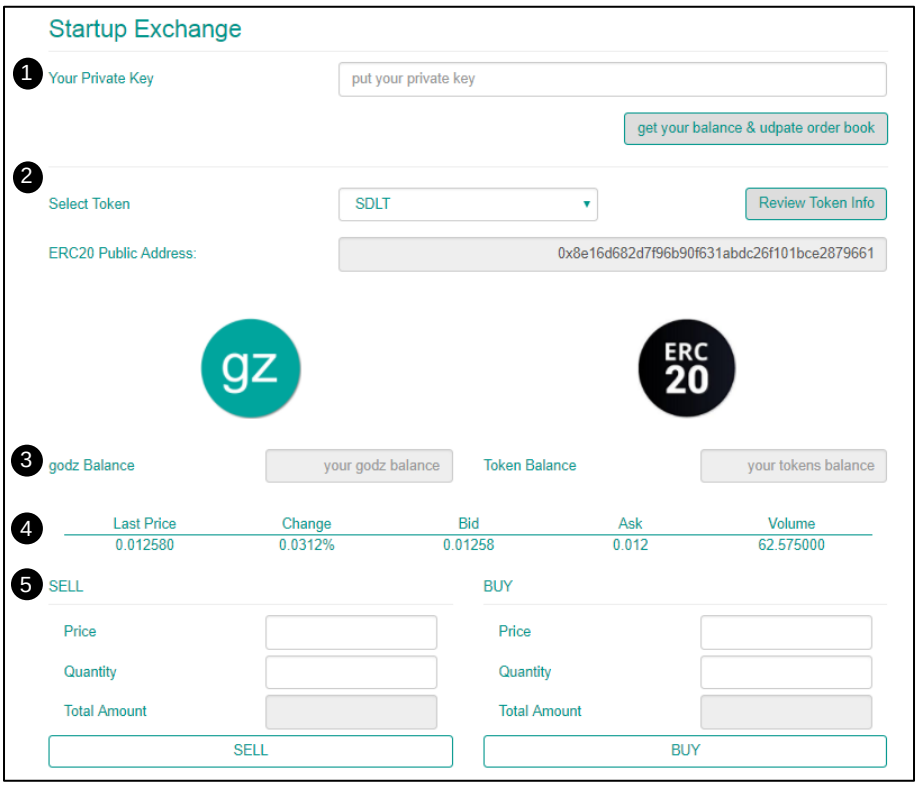
7) In this this final section, investor can review the last 5 trades executed in this exchange, with the date and time, type of trade (a buy or sell), the price, quantity and the amount of the trade involved.



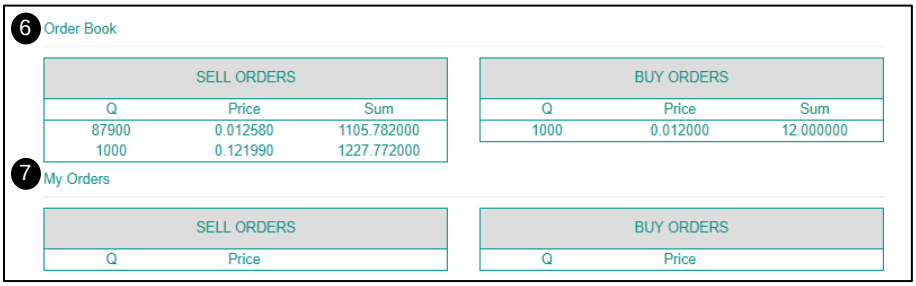
All rights reserved for mifuturofinanciero.com SpA

6.3.2 Startups

1) The investor needs to put the private key of his wallet here, in order to activate this section of the DApp.
2) The investor can select from the list the startup token that he wants to trade. By selecting the token, he will see the ERC20 Public Address of the token and also can click on the button of "Review Token Info" if he wants to know more about it.
3) Once that investor puts in his private key, the screen will show his balance of GODZ and the token that he selected above. If it's his first time on Godzillion, the balance of Token will be 0.
4) Here the investor can review the information of the last trade of GODZ for Token (in this case SDLT) executed in this exchange.
5) If the investor wants to buy Tokens of SDLT (for this example), he needs to review the information of the last trade or the order book to put in his buy order. The investor needs to put the price at which he wants to buy, the quantity, and total amount of GODZ that he needs to place in the buy order. Then he can click the Buy button, and if the validation is ok, the order will be placed. In case if he wants to sell Tokens of SDLT, and he has Tokens in his wallet, he needs to do the same steps, but click on the SELL button.

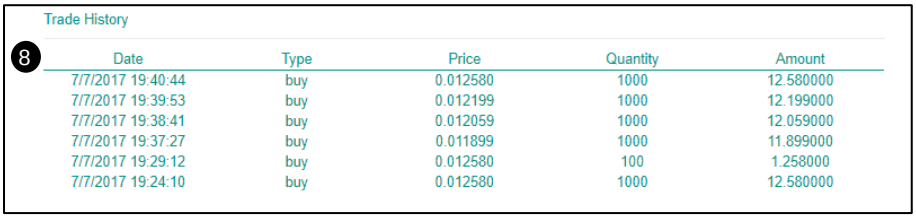


6) Here the investor can review the Order Book for the selected token specifically in this exchange, to analyze the Bid and Ask, to place his Buy or Sell order.
7) In this section, the investor can review the order that he has placed for this token specifically in this exchange, according to whether he wants to Sell the selected Token, or if he wants to Buy the selected Token. He can delete those orders if he wants because they have not been executed.



All rights reserved for mifuturofinanciero.com SpA

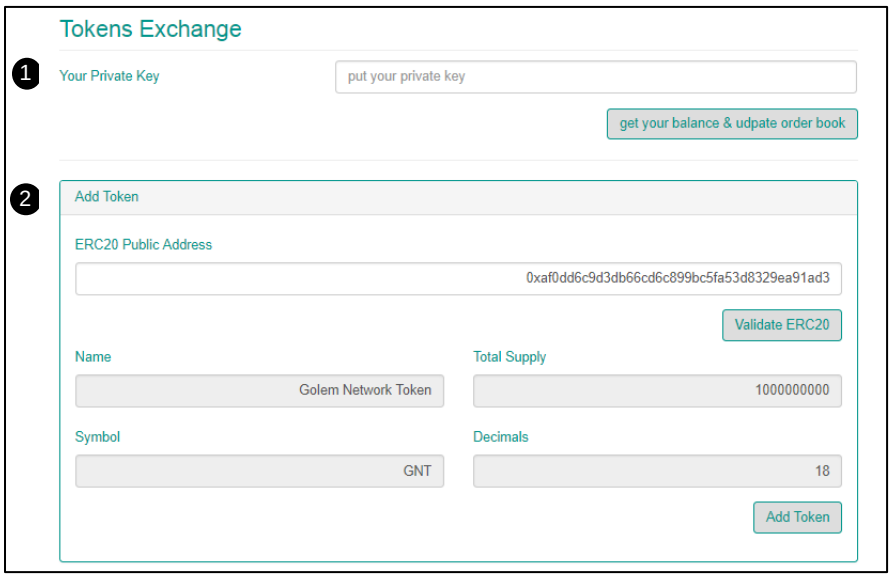
8) In this this final section, the investor can review the last 5 trades executed for the selected Token in this exchange, with the date and time, type of trade (a buy or sell), the price, quantity and the amount of the trade involved.



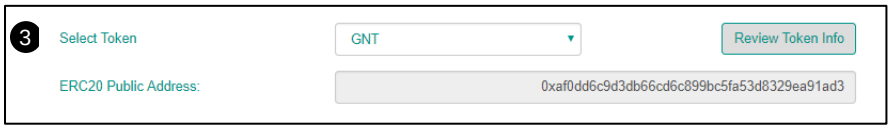
6.3.3 Ethereum Tokens

1) The investor needs to put the private key of his wallet here, in order to activate this section of the dapp

2) The investor can Add Tokens to this exchange, if he holds tokens from another exchange and he wants to transfer them here. In order to do that, he needs to put the public address of the ERC20 Token that he wants to add, and then click on “Validate ERC20” button. The information of that ERC20 Token will appear in the bottom, with the name, Symbol, Total Supply and Decimal associated with it. If the information is correct, the investor can click in “Add Token” and this exchange will allow this ERC20 Token to trade, listed on the exchange.

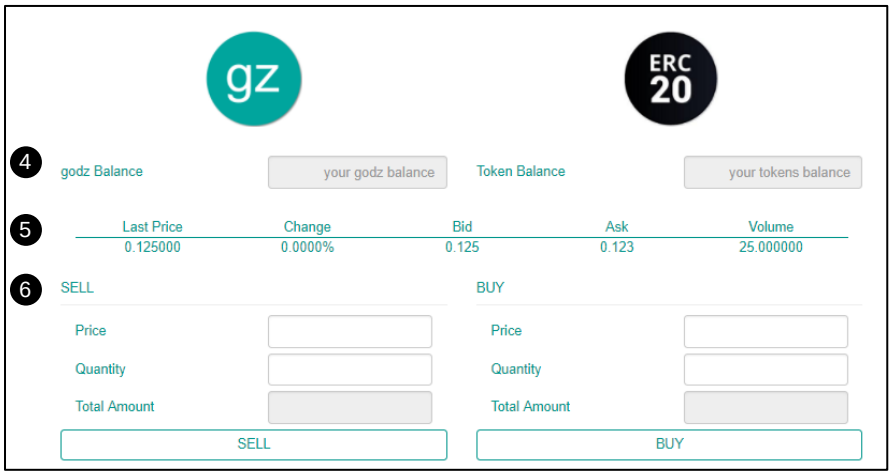


3) If the Token that the investor wants to trade is already listed on this exchange, he just needs to select it, and the ERC20 public address will appear, and also the investor can review information about this token by clicking in the “Review Token Info” button.

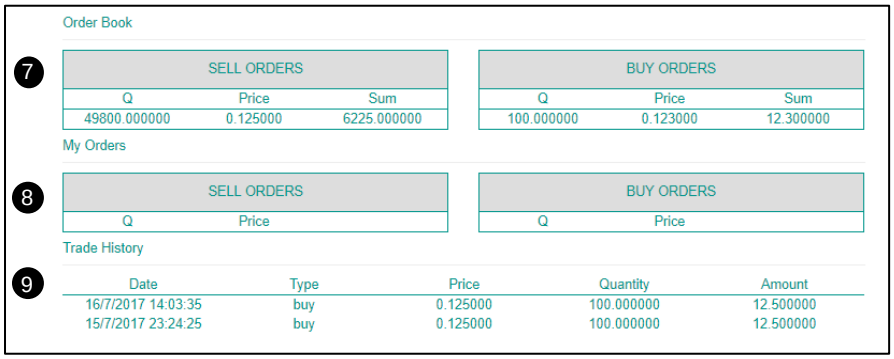


All rights reserved for mifuturofinanciero.com SpA

4) Once that investor puts in his private key, the screen will show his balance of GODZ and the token that he selected above. If it's his first time on Godzillion, the balance of Token will be 0.
5) Here the investor can review the information of the last trade of GODZ for Token (in this case GNT) executed on this exchange.
6) If the investor wants to buy Tokens of GNT (for this example), he needs to review the information of the last trade or the order book, to put his buy order. The investor needs to put the price at which he wants to buy, the quantity, and total amount of GODZ that he needs to place the buy order. He can then click the BUY button, and if the validation is ok, the order will be placed. In case if he wants to sell Tokens of GNT, and he have Tokens in his wallet, he need to do the same steps, but click on the SELL button.



7) Here the investor can review the Order Book for the selected token specifically in this exchange, to analyze the Bid and Ask, to place his Buy or Sell order.
8) In this section, the investor can review the orders that he has placed for this token specifically in this exchange, according if he wants to Sell the selected Token, or if he wants to Buy the selected Token. He can delete those orders if he wants.
9) In this this final section, the investor can review the last 5 trades executed in this for the selected Token in this exchange, with the date and time, type of trade (a buy or sell), the price, quantity and the amount of the trade involved.



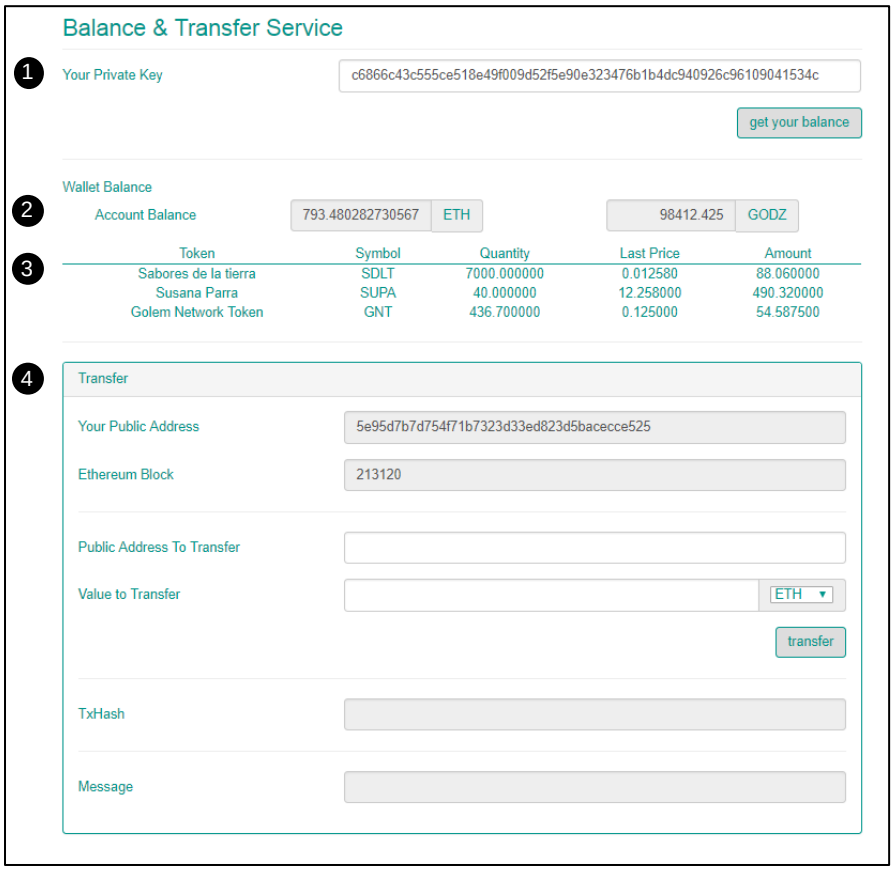
6.4 Transfers and History

6.4.1 Transfers

1) The investor needs to put the private key of his wallet here, in order to activate this section of the dapp.
2) In this section, the investor can review the balance of his wallet, of ETH and GODZ
3) In this section, the investor can review the balance of all the tokens that his wallet holds in this dapp, and can see the quantity that he or she has, the last price of his token on the exchange, and the valuation (in GODZ) of his tokens.

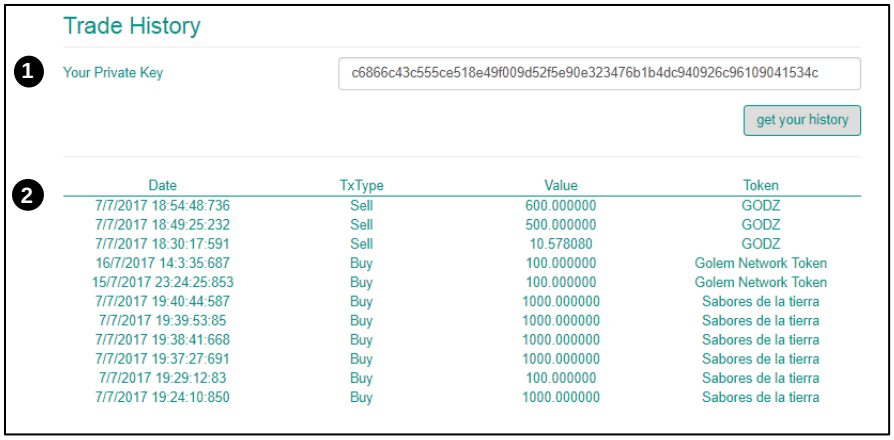
All rights reserved for mifuturofinanciero.com SpA

4) In this final section, the investor can transfer any of his tokens, GODZ or ETH to any other wallet in the Ethereum Blockchain. Selecting the type of value that the investor wants to transfer, the investor just needs to put the public address of the wallet that he wants to transfer to, put in the value to be transferred, and then click on the “transfer” button.



6.4.2Trade History

1) The investor needs to put the private key of his wallet here, in order to activate this section of the dapp.
2) In this section, the investor can review all the history of his trades from the different exchanges that compose this dapp. In this example, we can review the trades of this wallet, where we can see the date and time of the trade, the type of trade (if is a Sell or Buy), the value involved and the token traded.



All rights reserved for mifuturofinanciero.com SpA

7. Token Crowdfunding Campaign and the use of proceeds

300,000,000 GODZ will be created (Tokens ERC20) at Godzillion's Token Crowd Sale (TCS). People will be able to buy 210,000,000 GODZ with ETH. The process will be executed with a Swap Smart Contract, receiving ETH and delivering GODZ to the ETH sender. Purchasers will send ETH to the Godzillion Swap Smart Contract in order to receive GODZ. By doing so they will receive Godzillion Tokens (GODZ) at the rate of 160 GODZ per 1 ETH. The details are here below:

	Swap rate
GODZ for ETH	160
ETH for GODZ	0,00625

A participant must send ether to the Smart Contract account after the start of the crowdfunding period (specified as a block number). Crowdfunding ends when the end block is created, or when the amount of ether sent to the account reaches the maximum. The crowdfunding address will be announced at the crowdfunding start through the following official channels:

- Project webpage: www.godzillion.io
- Official Twitter: https://twitter.com/godzillion_io

Please, double-check the address before sending ETH. For security reasons, we advise to confirm the address using at least two different sources above. On the project webpage, you will also find a detailed guide on how to participate in the crowdfunding using the Ethereum Wallet.

Crowd Sale is implemented as a smart contract with a few simple parameters:

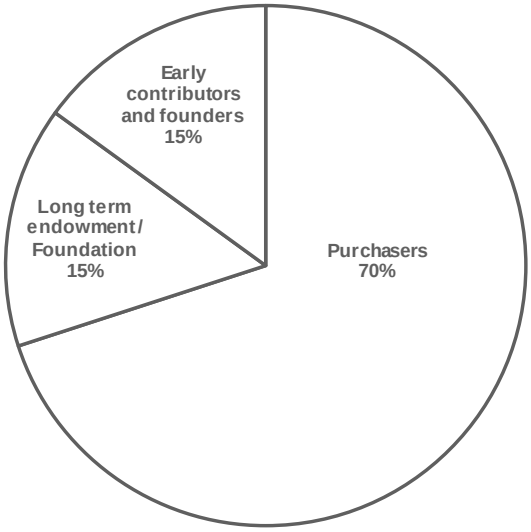
- Godzillion controls the contract and the address to which gathered ETH will be sent
- Percent of pre-allocated tokens is 30% (15% - Godzillion Team and Founders, 15% - Godzillion Foundation)
- StartBlock, EndBlock: these block numbers indicate the start and the end of the crowdfunding process

Another pool of 45,000,000 GODZ will be directly allocated to early contributors who have worked to develop the ideas, implementations, and supporting structures of the Godzillion project up to the time of the Crowd Sale. The size of this pool is 15% of the Total Quantity of GODZ. A second pool of 45,000,000 GODZ will be directly allocated to the Godzillion Foundation for payment of future expenses to be determined in its sole discretion. The size of this last pool is 15% of the Total Quantity of GODZ.

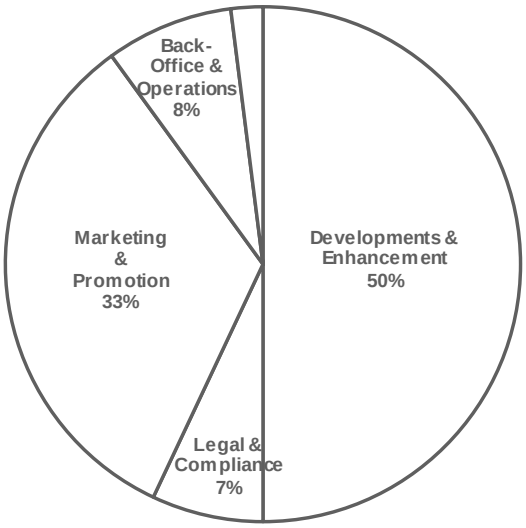
So, the total issuance of GODZ (300,000,000) is distributed as follows: 70% to Purchasers, 15% to a Long-Term Endowment for the Godzillion Foundation and 15% to the Early Contributors, Founders and the Team. As the road map explain above, the intended use of revenue is distributed as follows: 50% to Development & Enhancement, 8% to Back-office & Operations, 33% to Marketing & Promotion, 7% to Legal & Compliance and 2% in Various & Others.

The details are here below:

Token distribution



Intended use of revenue



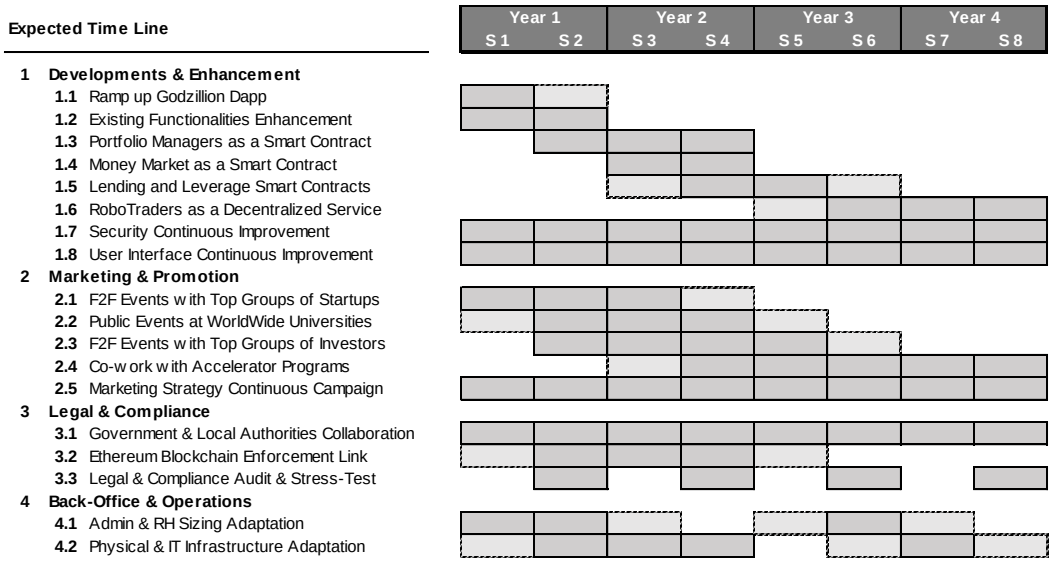
All rights reserved for mifuturofinanciero.com SpA

8. Road Map

Godzillion DApp is already operating and has already facilitated the financing, via issuance of tokens, of four Chilean startup companies. Trading in those tokens commenced immediately after the auction and issuance procedures were completed. In order to achieve the future milestones, a road map has been established. The road map encompasses four main branches of implementing priorities over the four-year planning horizon. The priority activities and enhancements include specific activities to achieve Development and Enhancement, Marketing and Promotion, and Legal and Compliance tasks and milestones.

- **Development & Enhancement**: This first year will start with the ramp up of Godzillion Dapp operations and enhancing the already operating functionalities as the front running alert. The second half of the year, a major task is to develop a Smart Contract that be able to implement portfolio management decisions, with settings and controls under the supervision of a human manager or with automated features. In the second year, a key task is to develop a Smart Contract that can manage functionality of a money market, connecting short term lenders with liquidity options with tokens as collaterals. The third year a major objective is to build a Smart Contract that can implement a Lending logic feature, which will oversee and implement a bond market for individuals and startup businesses. As the team learns from rolling out the above enhancements and features, the capstone projected development for the fourth year is a Decentralized RoboTrader with settings that can be controlled by Investors and a second type of RoboTrader with a specific market maker role. A constant theme during this four-year horizon will be improvement in Security issues as an overarching priority, and improvements in the User Interface experience.
- **Marketing & Promotion**: Our outreach to Startups groups will increase, so that Startups will launch more issuances of equity-linked securities, using the Godzillion Decentralized Application. As soon as the GODZ ICO ends, investors will hold GODZ and will be waiting for Startups to post their proposals to be voted on, and investors will be waiting to bid in the auctions in which the Startups seek to issue tokens. The second six months the outreach to Universities and their Incubators will be increase, in order to propagate the use of the Godzillion Dapp as a catalyst that brings together voters and successful potential startups. And, as Startups post more of their proposals, and more obtain financing, and as secondary market trading increases, Public Events will be help with Groups of Investors in order to explain the role of GODZ, the voting and issuance process, and the access to a secondary market based on Blockchain. After achieving those initial milestones, the second year the focus will involve collaboration with Accelerator Programs in order to improve information releases and startups formation processes.
- **Legal & Compliance**: In order to implement the outlined plan in a legally correct way, collaboration with different Regulators' Offices in different countries will be an ongoing effort. With the help of local Lawyers in each country, it is expected that the Godzillion Dapp will operate in all facets of the crowdfunding spectrum, with Regulators and Crowdfunding Community as well. During the second six months, with the help of Lawyers, the Team will implement a solution for linking the notarial issuance process with registration issuance into the Blockchain. As an ongoing financial reporting and Compliance policy, the Godzillion Project will be audited on an annual basis, and the auditor's reports will be posted in accord with standard procedures.
- **Back-Office & Operations**: After the Crowd Sale, the Team plans to start recruiting qualified people in order to be maintain technical and human implementation capacity. New hires will be mostly in the areas of Development and Marketing. The second semester it is also planned to relocate the Headquarters from Santiago to London in order to bring the Project closer to the European market and to deliver service to the many countries that are near that financial hub.

The detailed activities that are scheduled to be put into operation during the time span is illustrated in the Time Line diagram here below:



All rights reserved for mifuturofinanciero.com SpA

9. Corporate History

MiFuturoFinanciero SpA is a Startup co-founded by Rodrigo Sainz and Cristóbal Pereira, both Chilean entrepreneurs. The company's mission is to bring finance closer to people, creating fintech products and services to people to achieve their wealth creation process. From their earliest visioning sessions, their objective has been how to automate processes in the financial sector so that people could enter the financial market efficiently, without too much intermediaries in the process.

The first problem they sought to overcome was the friction that people face as they seek to access the financial market. In order to reduce that friction, in 2013 the company developed a web platform that allows people to review different types of financial products (from fixed income to alternative assets) in just one site. After launching the platform and attracting users to it for a year, they realized that there is a second barrier for people in order to have access to the stock market or even to the bond market. They need to have an initial equity of at least US $ 1,000 to use the full features of the platform. That was the minimum amount of equity that the users needed to have in order to open an account with the stock brokerage firm to have access to market and implement orders.

In order to resolve this friction, in 2015, they developed an App that helps people to round up their daily expenses, so the decision of spending or saving is no longer necessary. With this App people spend and save at the same time, so it is easy for them to channel their small amounts of savings into money market funds and accumulate US $ 1,000 in a short period of time.

At the end of 2015 the co-founders saw the potential of the Blockchain Technology in reducing friction in financial processes, decentralizing transfers and record keeping. After the Ethereum Project and the Smart Contracts logic was launched, they start to realize the enormous potential that blockchain will accomplish for trading and corporate finance. At the moment that the team were learning about this technology they immediately saw its suitability, and began to program their first smart contract on Solidity and deploy it on the Ethereum Blockchain.

In the first quarter of 2016 they coded their first smart contract to manage the issuance of bonds for companies and individuals, with a primary market and a private secondary market linked together. After learning from that experience, the founders decided to get 100% into blockchain technology, leaving behind the centralized logic, and they started to build a fully decentralized platform as a Smart Contract on the Ethereum Blockchain, the result is Godzillion.

The Godzillion Team work together since 2013 and they are:

- **Cristobal Pereira**, co-founder and CEO, holds a Master in Administration and an Engineering degree from UDD, Chile.
- **Rodrigo Sainz**, co-founder and CIO, holds a Master in Finance and an Engineering degree from UDD, Chile.
- **Eduardo Portugues**, CTO, holds a Computational Engineering degree from DUOC, Chile.
- **Matias Pereira**, Software Engineer, holds a Master in Administration and Engineering degree from UDD, Chile

Today the company's focus is to create products and services designed and programed as decentralized platforms that work on blockchain. Mifuturofinanciero.com SpA has received favorable press reports and coverage on radio and TV, due the innovation in the Chilean financial sector for platforms like mimolido and godzillion, the two platforms that are operating at this moment.

Shareholders and Advisor of the Board are prominent members of the Chilean and International financial community, with academic degrees, professional certifications, and extensive line management experience in traditional Chilean financial intermediaries.

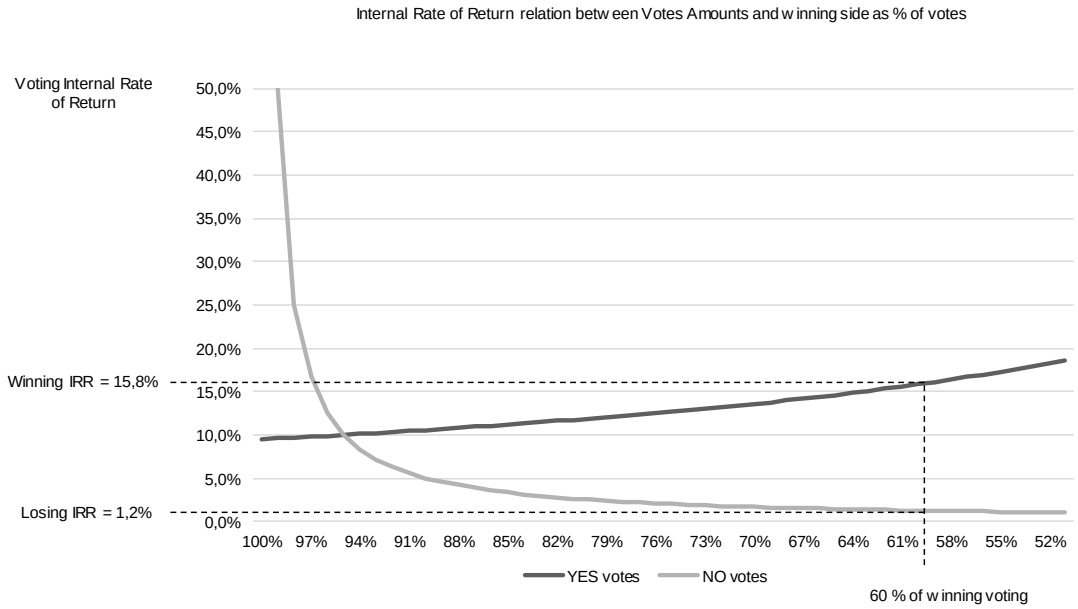
The Advisors of the Board are:

- **John C. Edmunds**, holds a D.B.A. in International Business from Harvard Business School, an M.B.A. in Finance and Quantitative Methods with honors from Boston University, an M.A. in Economics from Northeastern University, and an A.B. in Economics cum laude from Harvard.
- **José L. Ruiz**, holds a PhD in Managerial Science and Applied Economics from Wharton Business School, University of Pennsylvania, and a Master in Finance and Master in Economics, both from Universidad de Chile.
- **David Diaz**, holds a PhD in Business Intelligence, University of Manchester Business School, and a Master in Finance from Universidad de Chile.
- **Roberto Darrigrandi**, holds a PhD in Business Management from University of Liverpool and a MBA from Booth School of Business, University of Chicago.

All rights reserved for mifuturofinanciero.com SpA

10. Appendix

10.1 Voting Reward Structure, Internal Rate of Return:



10.2 Cost Structure, Gas needs:

The fee schedule G is a tuple of 31 scalar values corresponding to the relative costs, in gas, of a number of abstract operations that a transaction may effect.

Name	Value	Description
Gzero	-	Nothing paid for operations of the set Wzero.
Gjumpdest	1	Paid for a JUMPDEST operation.
Gbase	2	Amount of gas to pay for operations of the set Wbase.
Gverylow	3	Amount of gas to pay for operations of the set Wverylow.
Gmemory	3	Paid for every additional word when expanding memory.
Gcopy	3	Partial payment for *COPY operations, multiplied by words copied, rounded up.
Gtxdatazero	4	Paid for every zero byte of data or code for a transaction.
Glow	5	Amount of gas to pay for operations of the set Wlow.
Gsha3word	6	Paid for each word (rounded up) for input data to a SHA3 operation.
Gmid	8	Amount of gas to pay for operations of the set Wmid.
Glogdata	8	Paid for each byte in a LOG operation's data.
Ghigh	10	Amount of gas to pay for operations of the set Whigh.
Gexp	10	Partial payment for an EXP operation.
Gexpbyte	10	Partial payment when multiplied by dlog256(exponent)e for the EXP operation.
Gblockhash	20	Payment for BLOCKHASH operation.
Gsha3	30	Paid for each SHA3 operation.
Gtxdatanonzero	68	Paid for every non-zero byte of data or code for a transaction.
Gsload	200	Paid for a SLOAD operation.
Gcodedeposit	200	Paid per byte for a CREATE operation to succeed in placing code into state.
Glog	375	Partial payment for a LOG operation.
Glogtopic	375	Paid for each topic of a LOG operation.
Gbalance	400	Amount of gas to pay for a BALANCE operation.
Gextcode	700	Amount of gas to pay for operations of the set Wextcode.
Gcall	700	Paid for a CALL operation.
Gcallstipend	2.300	A stipend for the called contract subtracted from Gcallvalue for a non-zero value transfer.
Gsreset	5.000	Paid for an SSTORE operation when the storage value's zeroness remains unchanged or is set to zero.
Gsuicide	5.000	Amount of gas to pay for a SUICIDE operation.
Gcallvalue	9.000	Paid for a non-zero value transfer as part of the CALL operation.
Rsclear	15.000	Refund given (added into refund counter) when the storage value is set to zero from non-zero.
Gsset	20.000	Paid for an SSTORE operation when the storage value is set to non-zero from zero.
Gtransaction	21.000	Paid for every transaction.
Rsuicide	24.000	Refund given (added into refund counter) for suiciding an account.
Gnewaccount	25.000	Paid for a CALL or SUICIDE operation which creates an account.
Gcreate	32.000	Paid for a CREATE operation.
Gtxcreate	32.000	Paid by all contract-creating transactions after the Homestead transition.

All rights reserved for mifuturofinanciero.com SpA