

## About document

This document is a submission for contest «Informal specifications for DGO main service» ([link](#)).

The document contains:

- terms and definitions used in the subject area;
- a short description of the subject area;
- description of the purpose and goals of the System;
- a short description of the proposed functionality of the System;
- high-level description of the proposed functional architecture;
- description of the role model of the System;
- description of proposals for the target organization of the automated activity;
- description and / or declaration of specific practices and algorithms.

The document not contains:

- a complete set of policies and principles for organizing decentralized governance in the Free TON project:
  - in the absence of an agreed and / or widely accepted and / or approved set of policies and principles for decentralized governance (other than the very high-level principles described in the [declaration of decentralization](#));
  - due to the absence of this list in the [proposal for the creation of DGO subgovernance](#);
  - nevertheless, as a justification for certain proposals on the functionality of the System, the author refers to certain principles of organizing decentralized management, which seem obvious and / or correct to him;
- detailed and complete descriptions of entities, detailed descriptions of algorithms, complete lists of attributes of entities, etc:
  - since this document, in terms of its place in the chain of artifacts created during the design of the System, is part of the draft design;
  - since it makes sense to create complete specifications only at the stage of creating a working project (after the high-level definition of the concept, scope and high-level functional architecture of the System).

Remarks:

- a significant part of this document (especially the terminological base) was formulated as a result of studying the following resources:
  - <https://gov-playground.rsquad.io/>
  - <https://easy-vote.rsquad.io/>
  - <https://forum.freeton.org/>
  - <https://gov.freeton.org/>

# Terms and Definitions

**System** — an automated governance system DAO Free TON.

**Subsystem** — structural element of the System; a named entity that has an independent meaning as part of the System.

**Module** — structural element of the Subsystem; a named entity that has an independent meaning as part of a Subsystem.

**Proposal** — type of information block placed by the subject of the project in the "Portal" subsystem; proposal of the subject of the project to organize interaction with other subjects of the project in a way different from the current order (proposal for change).

**Post** — type of information block placed by the subject of the project in the subsystem "Communication platform" of the System.

**Submission** — type of information block placed by the subject of the project in the "Portal" subsystem, marked in a special way (as an application for a change).

**Contest** — type of information block, placed by the subject of the project in the subsystem "Portal", marked in a special way (as a competition).

**Group** — an isolated and named set of project subjects, which can enter into relationships with other project subjects.

**Tag** — property of the information block placed by the subject of the project in the subsystem "Portal", which allows grouping and managing the display of information blocks.

**User** — System user.

**On-chain part** — structural elements and data of the System, the placement and processing of which takes place in the Free TON blockchain.

**Off-chain part** — structural elements and data of the System, hosting and processing of which takes place outside the Free TON blockchain.

**Process** — a dedicated named activity carried out within the System.

**Group budget** — the amount of Free TON tokens that the group can dispose of, acting on its own behalf.

**Contest budget** — the amount of Free TON tokens that can be distributed to the winners of the Contest after its end.

**Voting** — an object type that can be associated with some objects of various types in the "Portal" subsystem to determine the final selection within the predefined conditions.

**Project** — Free TON

**Subject of the project** — a separate individual (project participant) or a group of project participants or a set of groups of project participants who can enter into relationships with each other and the project as a whole.

**DGO** — decentralized governance organization; in this document, DGO refers to the “automation domain” or “set of automation activities” depending on the context of the story.

## General regulations

### Short description of the subject area and problem statement

Organizing leadership and governance in a decentralized community is an important and complex task. The importance is ensured by the need for constant development of the project (which, obviously, will be significantly complicated in the absence of a decentralized governance system). The complexity of this task is provided by the lack of a single point of decision-making (which is characteristic of centralized systems) at the initial stage and the need to create and maintain, by definition, complex non-hierarchical network relationships between the subjects of the project.

Actually, the considered task is the establishing of effective governance of the decentralized community of the project by managing the relationship of the Project Subjects with each other and the Project as a whole.

The relationships (types of relationships) of the Subjects of the Project with each other and with the Project as a whole can be considered as the following set<sup>1</sup>:

1. **Economic interaction.** Interaction, the result of which is the transfer of project tokens between the subjects of the project (for example, the transfer of a reward to a participant in the competition for taking a prize).
2. **Changing the technical parameters of the project.** Interaction, the result of which is to change the parameters of the project (for example, an increase in the maximum number of network validators and / or a decision to use one or another software).
3. **System Changes.** Highlighted separately, but can be considered as a set of (1) and (2). At the same time, in the context of this paragraph, the term "System" is interpreted comprehensively, as a set of types of information systems support — in other words, a change in the guidelines and / or project policies will be considered in the context of a change in the System.

In other words, in a project, any relationship between the subjects of the project with each other and with the project as a whole can be typified as (1), (2) or (3), and only so. Then the task of developing the System at the top-level can be formulated as: “Develop the System within which the relationships between the subjects of the project of types (1), (2) and (3) in the subject area of the project will be formalized and automated”.

The main prerequisites for the development of the System are:

- the need to improve project management activities;

---

<sup>1</sup> Probably, the following set describes only the current instance of the governance system. At the same time, after the launch of shardchains, we will theoretically be able to talk about a set of instances of governance systems, each of which can govern the relationships of subjects according to its own predefined rules

- the need to introduce new management practices and policies;
- the need to create a convenient interface for interaction between Project Subjects;
- the need to create a socialized environment for interaction, exchange of experience, joint problem solving and implementation of mechanisms to stimulate Project Subjects to increase their activity in this area;
- the need to implement analytical tools for subsequent decision-making based on data analysis;
- the need to create additional and develop existing areas of interaction between the subjects of the project.

Thus, the tasks of developing the System are reduced to the following key areas of work:

- solving problems that hinder the work and reduce the efficiency of DGO — solves the need to optimize the functioning of the main management system of processes;
- creating a supportive environment for decision making — adds an additional level of functional value.

## **Purpose and goals of the System**

The purpose of the System is to organize the relationship of the Project Subjects between themselves and the Project as a whole in an effective way within the framework of the DGO principles and policies of the Project.

Goals and objectives of creating the System:

- minimize manual operations in the interaction of Project Subjects with each other;
- ensure the implementation of the principles and policies of the DGO project through their automation within the System;
- to provide the Project and the Subjects of the Project with a convenient automated tool that allows you to propose changes, discuss all aspects of the life of the Project, vote for certain proposals, control the implementation of changes within the framework of the selected types of relationships;
- provide the Project and the Subjects of the Project with a single point of accurate and consistent information about the Project;
- ensure the automation of economic management of givers, budgets of groups and competitions, other budgets of Project Subjects;
- create tools for analyzing the actions of Project Subjects;
- create tools for the participation of all Project Subjects in funding the budgets of other Project Subjects (crowdfunding);
- increased user satisfaction with a friendly interface.

## **Expected results**

1. A unified organizational and technical infrastructure has been created to implement the DGO principles and policies.
2. A complex of information, analytical and communication tools was created for decentralized Project governance.
3. Platforms and algorithms for related (non-core) DGO cases have been created.

## Scope and limits

The subject area of the Project is described in the section "Short description of the subject area and problem statement".

Limits:

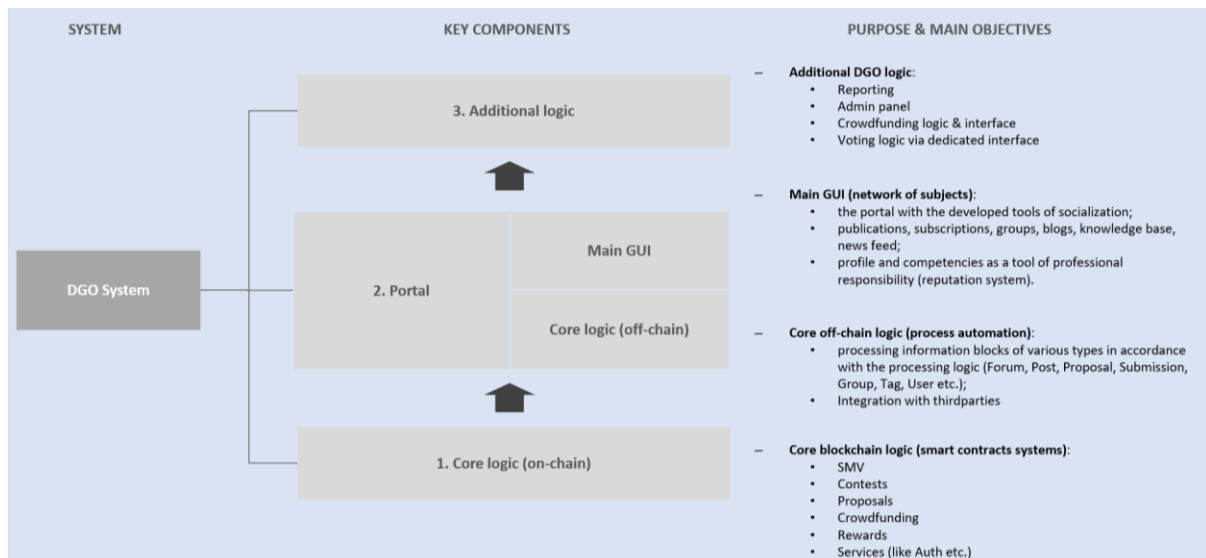
- at the stage of formation of this document, only conceptual solutions (conceptual and / or draft design) are proposed, which in the future must be transformed into detailed solutions during the development of the working design of the System;
- due to the large volume of tasks to be solved, the target functional architecture presented below in the text of the document will most likely not be achieved within a single iteration. This means that at the subsequent stages of the development of the System, a list of transitional functional architectures for the System will be formulated.

## System Concept

### The upper-level concept of the System

The system is an independent solution, architecturally related to the current DGO implementation and the Free TON blockchain as a bus and data source / storage.

The top-level target component architecture of the System is shown in the following figure.



*Figure 1 — Top-level target component architecture of the System*

The system consists of 3 interdependent subsystems:

1. **Core logic.** The subsystem is designed for storing and executing smart contract systems in order to ensure the functioning of the DGO process logic.

2. **Portal.** Communication subsystem of the Subjects of the Project with each other and the Project as a whole. This subsystem consists of modules: the first module is responsible for the user interface; the second module is responsible for the off-chain logic of the System.
3. **Additional logic.** The subsystem is designed to automate additional scenarios for using the System. These scenarios are very important for the organization of mature decentralized management: multi-format reporting on DGO operation, DGO administration, crowdfunding portal, an additional interface for voting.

## The relationship of purpose, goals and objectives and results

The proposed architectural solution, which includes three key subsystems, will solve the required basic tasks, as well as bring a number of additional values. The relationship between goals and objectives with the purpose of the subsystems of the System is shown in the following figure.

PURPOSE	OBJECTIVES	Core Logic (on-chain)	Main GUI Core Logic (off-chain)	Additional Logic	OUTCOMES
Organize the relationship of Project Subjects between themselves and the Project as a whole in an effective way within the framework of the Project's DGO principles and policies	1. minimize manual operations when Project Subjects interact with each other	+	+	+	1. A unified organizational and technical infrastructure has been created to implement DGO principles and policies
	2. ensure implementation of the Project's DGO principles and policies by automating them within the System	+			
	3. provide the project and project subjects with a convenient automated tool that allows them to propose changes, discuss all aspects of the project's life, vote for certain proposals, and monitor the implementation of changes within the selected types of relationships	+	+		
	4. provide the Project and Project Subjects with a single point of accurate and consistent information about the Project		+	+	2. Created the complex information and analytical and communications tools for decentralized governance of the Project.
	5. provide automation of economic management of givers, budgets of groups and contests, and other budgets of Project Subjects	+			
	6. create tools for analyzing the actions of Project Subjects			+	3. Platforms and algorithms for related (non-core) DGO cases have been created
	7. creating tools for involving all Project Subjects in funding the budgets of other project subjects (crowdfunding)			+	
	8. increase user satisfaction with a friendly interface		+		



- Performance
- Smart interface
- Data-driven decision making
- Full transparency
- Automation
- Scoring of Project Subjects

Figure 2 — Implementation of the Project tasks in the System

## Short description of the proposed functionality of the System

The following list shows the key functionality of subsystems, modules and components of the System:

1. **Core logic (on-chain).** The subsystem is intended for storing and executing smart contract systems in order to ensure the functioning of the DGO process logic.
  - a. **SMV module.** A smart contract system that implements the Soft Majority Voting logic.
  - b. **Proposal module.** A smart contract system that implements the logic of working with Proposals.
  - c. **Contest module.** A smart contract system that implements the logic of working with Contests.
  - d. **Rewards module.** A smart contract system that implements the logic of automatic payment of rewards.
  - e. **Crowdfunding module.** A system of smart contracts that implements the logic of collective funding of the budgets of the project subjects.

2. **Portal.** Communication subsystem of the subjects of the project with each other and the project as a whole. This subsystem consists of modules:
  - a. **Main GUI module:**
    - i. Forum
    - ii. News feed
    - iii. Filtering functionality and various ways to visualize information
    - iv. Subscriptions
    - v. Events and activities calendar
    - vi. Resource directories, reference books, discussion topics, etc.
    - vii. Authentication interface
    - viii. Interface for interaction with the Free TON blockchain
    - ix. User's personal account
    - x. User interface
    - xi. Tagging interface
    - xii. Post interface
    - xiii. The interface for working with comments
    - xiv. Voting interface
    - xv. Contest interface
    - xvi. Submissions interface
    - xvii. Group interface
    - xviii. Reputation management interface for project subjects
    - xix. Calling interfaces of the Additional logic subsystem
  - b. **Core logic module (off-chain).**
    - i. Logic of working with off-chain objects of the system
    - ii. Storing information about off-chain System objects
3. **Additional logic.** The subsystem is designed to automate additional scenarios for using the System. These scenarios are very important for the organization of mature decentralized governance: multi-format reporting on DGO operation, DGO administration, crowdfunding portal, additional voting interface:
  - a. **Admin Panel module**
    - i. Off-chain moderation
    - ii. Backup
    - iii. Admin Tools
    - iv. Reference support for parameterized templates:
      - types of Votings;
      - Proposal types;
      - Rewards types;
  - b. **Reporting & Analytics module**
    - i. Various analytical tools and reports available to project actors
  - c. **Crowdfunding Platform module**
    - i. Additional interface for working with crowdfunding
  - d. **Voting module**
    - i. Additional interface for convenient voting

This list of functionality should be revised during the detailed design of the System.

## Role Model of the System

The role model and rules of access and administration within the System are built taking into account the peculiarities of the DGO subject area and the proposed component, functional and logical-technical architectures of the System.

It is assumed that the role model of the System will be incremental. This means that each user of the System will be assigned a primary role by default ("User"), and additional functionality will be available to the user after adding additional roles. In addition to the incremental role model, the user can also have relative roles in accordance with the restrictions imposed by the DGO activity (for example, the participation of the user-subject of the Project in a subgovernance or in a group of experts when voting).

This section summarizes the key functions within the structural elements of the System for large user types:

1. System User — the main role assigned to any System user;
2. Content Administrator — an additional role that allows you to perform certain administrative functions with the content of the System;
3. System Administrator — an additional role that allows you to perform certain administrative functions within the System;
4. Jury — a relative role, a special case of a user (1) is used when it is necessary to limit the number of voters (for example, for submission) within a Group;
5. Validator is a relative role, a special case of the user (4) is used to resolve some issues for the type of relationships between subjects (2) (described in the section "Brief description of the subject area and problem statement").

The functions of the System, which are performed automatically, are also indicated, but refer to any type of user. In this case, in the general case, we can say that a user of type (1) has access to view any content posted in the System (subject to the process restrictions specified in the section "Proposals for organizing automated activities").

Users of types (4) and (5) are System users with relative roles. This means that the presence of these roles depends on the values of the parameters of the System object, for which users of types (4) or (5) must perform some specific actions. Example: to judge a competition for the development of smart contracts, a group of users "Smart Contract Experts" is involved. In this example, members of the Smart Contract Experts group receive special rights to work with bids for the above competition.

System structural elements	Functions	Roles				
		System Administrator	Content administrator	User	Jury	Validator
1. SMV module	View and execute logic					
1. Proposal module	View and execute logic					
1. Contest module	View and execute logic					
1. Rewards module	View and execute logic					



<b>1. Crowdfunding module</b>	View and execute logic					
<b>2. Main GUI module</b>	Forum					
	News feed			X		
	Filtering functionality and various ways to visualize information			X		
	Subscriptions			X		
	Events and activities calendar			X		
	Resource catalog			X		
	Authentication interface			X		
	Interface for interaction with the Free TON blockchain					
	User's personal account			X		
	User interface			X		
	Tagging interface			X		
	Post interface			X		
	The interface for working with comments			X		
	Voting interface			X	X	X
	Contest interface			X	X	X
	Submissions interface			X	X	X
	Group interface			X		
	Reputation management interface for project subjects			X		
	Calling interfaces of the Additional logic subsystem			X		
	Search			X		
<b>Core logic module (off-chain)</b>	Logic of working with off-chain objects of the system	X				
	Storing information about off-chain System objects	X				
<b>Admin Panel module</b>	Off-chain moderation		X			
	Backup	X				
	Admin Tools	X				
	Reference support for parameterized templates	X	X			

<b>Reporting &amp; Analytics module</b>	Various analytical tools and reports available to project actors			X		
<b>Crowdfunding Platform module</b>	Additional interface for working with crowdfunding			X		
<b>Voting module</b>	Additional interface for convenient voting			X	X	X

*Table 1 — Matrix of functions and roles*

Notes:

1. The main interface for user interaction with all subsystems (with all functionality) of the System is the “Portal” subsystem.
2. For the purposes of administration and maintenance, separate access points to the subsystems and modules of the System can be created.

## **High-level description of the proposed logical-technical architecture**

The top-level diagram of the logical-technical architecture is shown in the following figure.

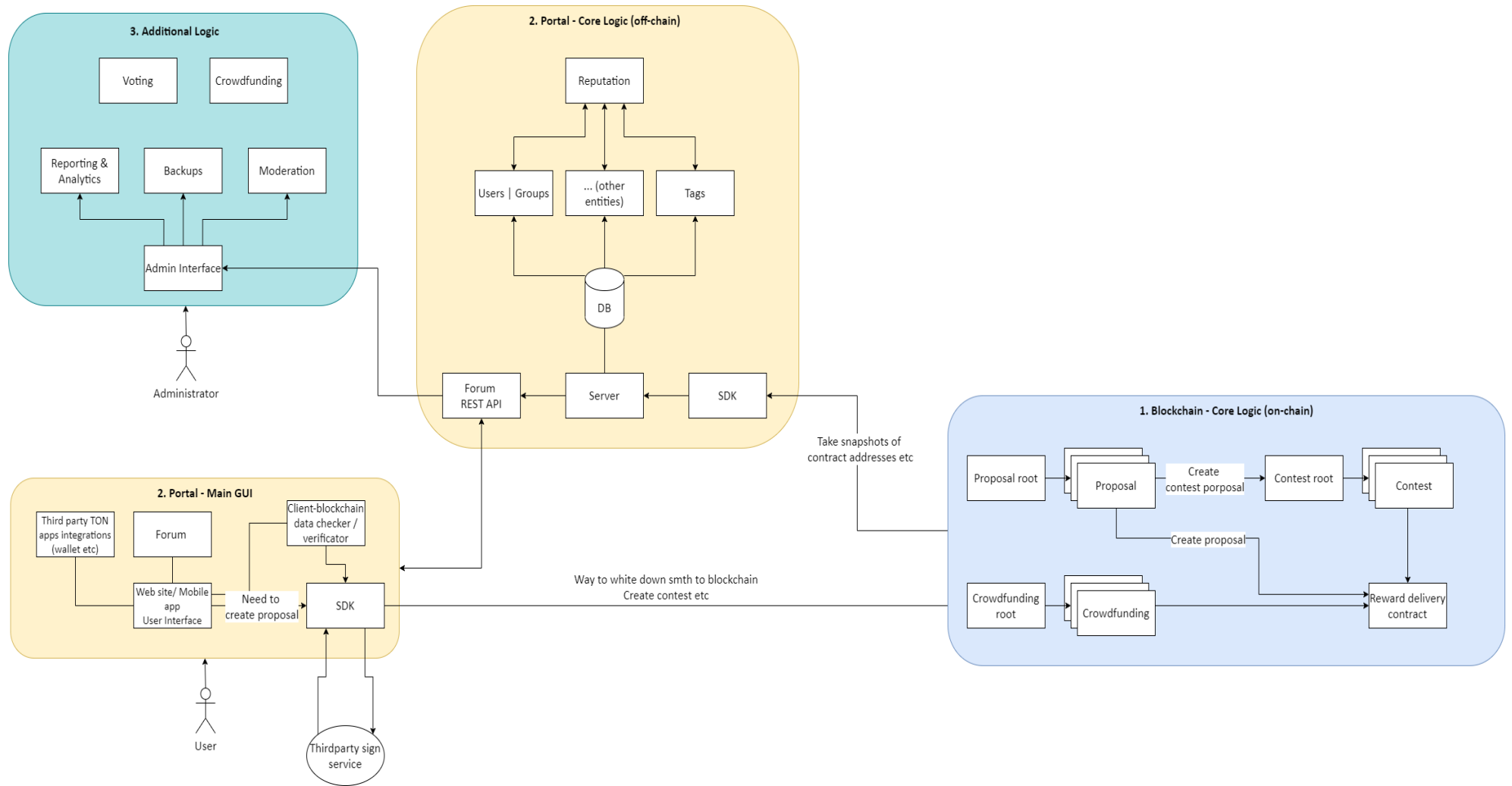


Figure 3 — Top-level architecture of the System

The top-level architectural scheme reflects:

- subsystems within the System;
- basic functionality of subsystems and modules;
- main interfaces of interaction between subsystems;
- the main events in which there is an interaction between the off-chain and on-chain parts of the System;
- main points of entry into the System for existing types of users.

## **Proposals for organizing automated activities**

### **Portal**

The Portal is the main official communication platform of the Project Subjects (in this document, we do not consider numerous chats). The System's Portal is an analogue of the current forum. However, there are some differences. They are as follows:

- additional functionality is available on the portal relative to the current functionality of the Forum: Project calendar, news feed;
- all DGO functions are available on the Portal, namely:
  - creation of Proposals;
  - creating Submissions;
  - voting for Proposals and Submissions;
- the above DGO functions, unlike the current gov.freeton.org implementation, are completely on-chain.

### **Authorization**

The Portal should support on-chain user authentication in order to ensure seamless integration of the System's functionality when the System user enters a seed phrase once.

### **Types of information blocks of the communication platform**

It is possible to create the following types of information blocks on the portal:

- Post;
- Proposal;
- Submission;
- Contest.

The life cycle of information blocks will be discussed below. The attribute model of information blocks should be revised during the detailed design.

### **Other types of information objects of the communication platform**

- Group;
- Tag;
- User.

## Post

An analogue of a post on the current forum or on a page in a social network. The post can be written in free form. The functionality of the communication platform should allow the user to beautifully design custom texts, attach links and attachments to the post.

Attribute composition (hereinafter, data types should be selected in accordance with the meaning of attributes and the nomenclature of possible values):

- Theme;
- Post body;
- Links and attachments;
- Author;
- Date and time of creation;
- Tags.

Post can be commented.

## Proposal

Any post can be converted to Proposal. In this case, this post must meet the design requirements for Proposals. Obviously, the layout of Proposal depends on the type and parameters of the Proposal. Accordingly, for each type of Proposal, the System must support a certain template, according to which the author will have to issue a Proposal. Templating is necessary to unify the information entering the System, as well as to uniform and simplify the automation of DGO processes. Proposal can be submitted on behalf of the group.

The task of compiling a catalog of Proposals types is of course the task of the working design phase of the System, but this Catalog will contain at least the following types of Proposals:

- Contest;
- Conducting a chain of Contests;
- Creation of subgovernance;
- Funding of the Subject of the Project — situations are possible when it will be necessary to organize the allocation of tokens upon request, for example:
  - to allocate a grant / advance for the development / implementation of something by the internal project group (however, such cases need to be minimized);
  - covering recurring generally useful expenses (advertising, encouraging performers of small works, etc.);
- Partnership;
- Changing the technical parameters of the Project;
- Changing DGO or System parameters;
- Choice of the jury (inclusion in a closed group);
- Creation of an application for fundraising (Crowdfunding);
- Proposal in "free form" (for starting work, the complete Catalog has not yet been developed).

From the point of view of the attribute composition, the Proposal should:

- determine the target action (for example, launching a competition, allocating funds, changing the network configuration, adding a tag to the System);
- set time limits, budget and quality of results (if applicable);
- set funding conditions (advance payment, payment after the competition, vesting, payment upon reaching KPI);
- determine the criteria for payments (for example, payments according to the occupied places or payments according to the average score);
- unambiguously determine the group of the jury for voting;
- unambiguously determine the parameters of voting (type of voting, voting parameters, for example, the total of votes for a positive assessment).

The Proposal type and its attribute composition should determine the cost of creating the Proposal. The parameters need to be determined during the detailed design of the System, but the basic proposal is:

- 1% of the requested budget is blocked;
- If the Proposal is accepted, the funds are returned to the author with a multiplier  $> 1$  (as a reward for creating the proposal needed by the project).
- If a refusal to accept the Proposal follows, then the blocked budget is not returned, but is used to encourage the voters.
- If no budget is requested, then a fixed cost is applied (depending on the type of Proposal).

In the user interface, Proposals are tagged with a special tag. Additionally, the tag marks the Proposal status.

Proposal statuses:

- New;
- Voting;
- Clarification;
- Accepted.

## **Voting**

The user can send a correctly drawn up Proposal for voting. The type and parameters of voting are determined by the Proposal or Submission parameters.

There are 2 types of voting according to the method of counting votes for making a final decision:

1. Voting based on quantification by the jury;
2. Voting based on the total of the jury's stakes.

The basic option is (2).

By the type of voting scale, the following types of voting are distinguished:

1. Yes / No / Abstained;
2. Scale / No / Abstained.

The basic option is (2).

Vote count and voting results (Contests with a large number of participants):

- The criterion for winning is the highest average score;
- Must vote at least the number of the jury (or the amount of the jury's stakes) determined by the voting template;
- “No” — equates to the lowest mark on the scale adopted in voting. The jury is considered to have voted;
- Equal amounts and / or number of juries — prize money is divided equally or "No".
- If more than 50% of the jury members (the amount of the stake) voted "No", then the proposal is automatically rejected.

## **Contest & Submission**

Contest is launched in case of a positive vote for Proposal. Contest is characterized by a budget. The Contest parameters are determined by the Proposal parameters.

Contest statuses:

- Upcoming;
- Underway;
- Voting;
- Ended.

Contest inherits tags from its parent Proposal. In this case, the service tags that indicate the Proposal statuses should be replaced with similar ones related to the Contest.

Applications for the Contest are submitted using Submissions. If no applications are submitted for the Contest, then it ends and the budget is returned. Submission is paid.

Based on the results of the completed Contest, payments are automatically initiated in accordance with the parameters of the contest.

## **Group**

Group — a dedicated named set of Project Subjects or users of the System.

Any user of the System can create a Group. Groups can be closed or open. Anyone can join an open group, a closed group can only be entered as a result of voting by the current members of the group.

## **Tags**

Tags define basic filters for content on the Portal. You can add from 1 to 5 tags to an information object, including automatically added service tags. At least 1 tag must be a non-service tag.

In addition to those existing on the Portal, you can create additional tags by filling out the corresponding Proposal. Technically, users with the System Administrator and Content Administrator roles can create a new tag.

During the detailed design, it is necessary to determine the color scheme of the tags used.

## Voting interface

The system should allow the user to vote using two interchangeable interfaces:

1. Voting directly from the Portal user interface;
2. Using a specialized interface for voting (analogous to gov.freeton.org).

There must be at least 2 supported voting interfaces, including one professional interface. This is necessary because a community of professional jury members is already being formed, who spend a significant amount of time evaluating something or voting for something. A dedicated voting user interface should be developed for this category of Project Subjects.

## Reputation

In the context of this System, it makes no sense to consider any model for calculating reputation other than the group one. This thesis can be justified by fixing several "reputation postulates":

1. The term "reputation" only refers to the expert reputation in the Project.
2. The expert never takes decisions concerning the objects of the System (Proposals, Submissions) alone. The consequences of this postulate:
  - a. there are no expert groups from a single project subject.
  - b. there is no personal reputation, but only a group one.

Reputation is measured in reputation units, which should relate to the sum of the budgets in which the group was a jury group. It is possible to arrange the automatic group selection jury for the judging of the competition or evaluation Proposal based on the value of "Reputation" of a group of judges (for example, automatic selection of the corresponding group of the jury with the maximum value of the indicator of "Reputation" - related Proposal Tags).

Developing requirements and a mathematical model for a group reputation indicator is a task of working design of the System, but at the current stage, we can formulate prerequisites for the formulation of the reputation model:

- the selection of a group for judging should be based on the value of the "Reputation" indicator;
- reputation should be considered for each tag-the direction of the group's expertise;
- calculation of reputation should be correlated with the amount of rewards in contests;
- the model should take into account reputation penalties for non-performance of jury duties in Contests and Proposals;
- the model should take into account the increase in group reputation with 100% voting (all members of the group voted);
- the model should take into account the reputation penalty when submitting an arbitration on the voting result;
- to prevent permanent hegemony of one group when evaluating works in any direction, it is necessary to calculate the "Reputation" indicator by analogy with the calculation of the rating in tennis (for tennis: the rating is calculated on the current date based on competitive results during the year from the current date; the rating takes into account the results of the previous period and increases in the case of the best sports results for the last year, decreases – in the opposite case). For example: the group currently has a rating of 1000 for Tag1. Let's assume



that we have a six-month cycle. In the last time period (3 months), the group increased its reputation by 200 (from 800 to 1000). Accordingly, within the next period, the group is guaranteed to "burn" 800 reputation points. At the same time, the group can get additional reputation points by participating in judging contests. Let's assume that a group participated in judging contests during the current time period and earned 300 reputation points. In this case, the total reputation of the group will be  $200 + 300 = 500$  points.

## **Additional functions**

### **Reporting & Analytics**

All data stored in the System (both on-chain and off-chain) should be used for creating reports and analyzing the activities of the DGO processes and the Project as a whole.

After starting the 1st queue of the System, you need to create requirements for the module "Reporting and Analytics".

### **Crowdfunding**

Collective funding of ideas and suggestions that arise in the community is an important topic. Obviously, this functionality should be implemented on the basis of a communication platform that is used by all active Project members.

After starting the 1st queue of the System, you need to create requirements for the "Crowdfunding" module.