Viite

# Model and build decentralized services on blockchain

# Viete.io

## Decentralized mass market services on the Blockchain

## Introduction

Companies that provide mass market services create a product or service with a certain level of quality and guarantees service that solves problems in the event of a defect.

The quality of service and the level of defects in the company are monitored. Technologies are selected and procedures are built to guarantee the required quality of service with a low level of rejects. The predicted low level of defects and warranty support in case of defects are included in the cost of the goods by slightly increasing it and by using the Queuing theory which is a reliability theory using statistical and probabilistic methods that can be used to predict the scrap rate and calculate the associated margin.

## Centralized Company

In a centralized company, the technology and procedures to achieve high quality and low rejects can be trade secrets. In this case, technologies, procedures and payment methods for them may be available to a limited number of employees. These people, being the beneficiaries of the company, control the level of quality and defects.

## Decentralized Company

In a decentralized company, all participants are the beneficiaries of its activities. Therefore, technologies and procedures aimed at achieving the desired level of quality, defects and related calculations must be public. This will allow the participants of the decentralized company to control the level of quality and defects. It will also allow people who want to join the company to be convinced of the reliability of its technologies and procedures.

## Format for Describing Technologies, Procedures and Calculations

A decentralized company needs a format to clearly and convincingly describe its technologies, procedures and calculations, which will prove that:

- The declared level of quality and defect is achievable in the company
- The cost of the warranty service is correctly calculated based on the assessment of the defect level
- The economics of the company converges, taking into account the cost of production, the amortized cost and the cost of the warranty service.

The format should include:

- Requirements for the level of quality and defect
- Descriptions of technologies and procedures
- Description of the calculation of the expected level of quality and defect
- Evidence that the described technologies and procedures, taking into account the calculation, meet the requirements for the level of quality and defect
- Calculating the convergence of the economy

Viete.io suggests using a visual language format (i.e. schematics/diagrams) based on category theory. Diagrams are often used to describe business procedures, supply or manufacturing chains. The theory of categories makes it possible to include formulas in these schemes, including calculations on the level of quality and defect, or economic calculations. In Viete.io, based on category theory, support for visual evidence is implemented, which visually allows you to verify the correctness of all technologies, procedures and calculations.

## Model-based Approach for Description

Description of technologies and procedures, requirements, calculations, and evidence should be clear and visual. In this case, the participants of the decentralized company will be able to effectively work with the description and interact with each other based on it. To achieve simplicity and clarity, Viete.io suggests using the same approaches that consumers use to describe complex, large, and intricate systems:

1. The user breaks the system into separate models, describes them individually, and then depicts how the models interact and relate to each other which provides the whole picture. The basic definitions of the composition and commutativity of category theory makes it possible to describe this approach in the form of diagrams. This mechanism is implemented in Viete.io.
2. The users can describe a simplified model of a complex system, and then add details to the model until it is fully described. The concept of a “functor” from category theory allows users to depict a simplified and complete model in the form of a diagram, comparing them with each other. This mechanism is also implemented in Viete.io.
   a. This mechanism adds an important element of control, since it allows users to check that the simplified model accurately matches the complete one, i.e. describes precisely its behavior at key points.
   b. It is convenient to start describing a complex system with a sketch – a principle (high-level) model that expresses the main ideas and then draws up a complete description of the system, referring to the principled one.

Viete.io offers several tools to simplify the model-based approach.

1. If something changes in the high-level model, then the changes can be propagated to the associated and more highly-detailed models. For example, if the business model changes, the changes can be propagated to the technical and financial models. Viete.io provides a smart refactoring tool for this distribution. Changes are carried along the lines of comparison between models as new elements and links are added or removed in the diagrams. The user sees these elements and connections and has the freedom to supplement them at the technical level.
2. It may be that a change at a technical level leads to a change in a higher-level business model. The comparability of models at different levels ensures that the need for an update is taken into account. Smart refactoring will allow users to deploy the changes and run them through all related models.
3. If a complex change is to be implemented within the company, it is convenient to describe the future operational model of the company and draw up a plan for the transition to new technologies and work procedures. Viete.io makes it convenient to draw up such a transition plan. Users can match current and new models, note the similarities and differences between them, and draw up a map for the sequential update of the current model to a new one.
   Such a plan of updates tethered to the original and the new models ensures that all stages of the transition are implemented logically and lead to results. The notion of natural transformation, along with composition and commutative diagrams from the category theory correspond to this general process. Viete.io also incorporates a host of visual tools for convenient operation with the models.

The editor hides terms from the category theory,

allowing non-specialists who are not familiar with the theory to work with it. For examples of using Viete.io, see the appendix below.

## Model-based Approach for Description

When calculating the expected level of quality and defect, probabilistic methods are used, which, based on the reliability of the work of the basic elements or members of the company, calculate the final characteristics for the entire company.

The level of reliability of the basic elements or participants is often not known in advance. In this case, it is calculated from the historical data of the company's functioning. Over time, with changes in technology or human behavior, the level of reliability may change and must be recalculated. It turns out that the calculation of the expected level of quality and defect needs to be regularly updated as collecting the data is necessary for the calculation.
The scheme of the company's functioning with a regular recalculation of the expected level of quality and defects is as follows:

1. Collecting big data about the work of elements or the behavior of participants
2. Calculation of the expected level of quality and defect
3. Comparison of calculation results with declared/required levels
   a. If the required level is not achieved, you can apply changes in technologies/procedures or changes in the declared/required levels of quality and defects
4. Recalculation of the company's economy taking into account the changed level of defect and checking the convergence of the economy
   a. If the economy converges, then you will work with an updated forecast
   b. If the economy does not converge, then you will have to change technologies procedures to achieve convergence of the economy

## Working with Blockchain

Various technologies can be used to implement the work of a decentralized company. The use of blockchain to store all key data allows achieving a high degree of trust, reliability and autonomy in the work of the company. All data related to the redistributionoffundsand othervaluescan be stored on the blockchain - this is a typical approach.
In this case, parts of technologies and procedures related to the redistribution of funds and other values are recorded in the form of smart contracts. You can additionally store big data on the blockchain, which is necessary for recalculation. In this case, the recalculation code can be implemented in the form of smart contracts, and changes in technologies and procedures in the event of an unsatisfactory expected defect rate can be carried out by using voting on the blockchain.

## Integration of Viete.io with the Blockchain

The technologies, procedures and calculations described in Viete.io using category theory are mathematically rigorous and unambiguous.
If we note which part of the description should be represented by smart contracts, then these smart contracts can be generated from the description. With a verified code generator, this approach will eliminate the appearance of errors in the implementation of technologies, procedures and calculations in the form of smart contracts.

Thus, Viete.io allows you to visually describe the work of a company, in one format presenting a description of technologies, procedures, calculations of the level of quality and defects, economic calculations and evidence of the correctness of technologies and procedures to be able to deploy the company's activities on the blockchain.

## An example of Implementation for a Grocery Delivery Company includes:

Consider, as an example, a decentralized fast food delivery system from restaurants. The delivery system guarantees that food will be delivered within 1 hour. Couriers are the beneficiaries of the system.

Couriers are notified of new orders, ride electric bicycles to the restaurant and after waiting a while for the food to be prepared, they receive the order and deliver it to the client.
Let's introduce the variables:

**t_g** — the time it takes for the courier to reach the restaurant
**t_w** — waiting time for order readiness
**t_c** — the time it takes for the courier to reach the client
**t_g + t_w + t_c <1** (1) — the condition that food will be delivered in 1 hour
**t_g + t_c <1 – P (t_w, 99%)** (2) — the condition that the courier with a probability of 99% will manage to deliver the order
**t_g + t_c <1 – P (t_w, 95%)** (3) — inequality

The waiting time for an order in a restaurant may depend on how busy the restaurant is with other orders. We can consider t_w as a random variable. If we want to guarantee delivery within 1 hour with a 99% probability, then we need to take the percentile P (t_w, 99%) to estimate the waiting time for the order.

Of course, the t_g and t_c at the time of receiving the order can also be calculated only with a certain accuracy. This can be done by either relying on ready-made services or using your own calculation methods. Nevertheless, if condition (2) is not met for any courier, then we will inform the user that there are no current free couriers, and we will not accept the order which results in potentially losing income and profit.

If we reduce the quality of service requirements and guarantee delivery with a probability of 95%, then instead of t_w we take P (t_w, 95%), which will be less than P (t_w, 95%), and we get the inequality (3). Inequality (3) will be fulfilled more often, and we will take more orders.

Another way to increase the number of orders is to predict the number of orders typically in a particular area for each hour.
In that case, we can send couriers from areas with less activity to areas that are in high demand.
For example, if there are more couriers in an area than the number of expected orders then we can direct the couriers to a place where there are fewer couriers and would be more beneficial to that area.
The expected number of orders will be dependent on the area, time and even the weather which can be obtained based on historical data.

There is also a possibility of cases where the courier's electric bike breaks down which would result in the courier not being able to deliver the order. In such cases, it can be attributed to defects. To prevent this, we will build a model of reliable electric bicycles and by carrying out diagnostics on time which reduces the number of cases of defects to a minimum and control the amount of defects.

This example demonstrates the relationship between the level of quality/defects and the economy of a decentralized company. It also demonstrates the methods of probability theory and machine learning that can be used to describe technologies and procedures within a company.

## A Framework with Methods of Probability Theory and Machine Learning

Typical methods of probability theory and machine learning can be assembled into a framework that will simplify the creation of decentralized

companies by simplifying the selection of technologies and drafting procedures that ensure a high level of quality and convergence of the economy.
The basic framework can be supplemented by creating a repository for open-source descriptions in the Viete.io language. The most stable and used solutions from the repository can be transferred to the framework.

## Updating the Economic Model of the Company

Market conditions can change over time, reducing the company's bottom line. In this case, it becomes necessary to update technologies or procedures within the company to restore economic efficiency.

**Example:**

So, in the example above, the waiting time for the order **t_w** may increase. For example, the number of visitors in restaurants will increase, and as a result, the average time to prepare orders for all customers, including couriers, will increase. In this case, the distribution of **P (t_w, 99%)** in the inequality **t_g + t_c <1 – P (t_w, 99%)** (2) so that the inequality is less likely to hold. As a result, couriers will accept fewer orders. The turnover and profit of the company will decrease.

Under these conditions, it is possible to replace the percentile **P (t_w, 99%)** with **P (t_w, 95%)**, which will lead to an increase in the level of rejects (orders that are delivered with an excess of the promised client waiting time of 1 hour) or you can start displaying orders with a waiting time of more than 1 hour in the client application interface, allowing you to choose. Also, you can offer customers to place orders with an indication of the exact delivery time no earlier than 2 hours from the moment of ordering.

All of these options affect the number of orders accepted, courier loading, customer satisfaction and customer behavior.

Based on historical data, it is not always easy or possible to simulate the behavior of clients and the work of a company under new conditions. Whenever possible, modeling requires careful interpretation of historical data and correct hypotheses.

## Updating the Economic Model of the Company

AA company may consider several plans with different changes in technology/procedures and economic models. The adoption of one of the plans will entail a change in smart contracts, which spell out the company's procedures on the blockchain.

We need a procedure for changing smart contracts, which will be controlled by all participants in a decentralized company. Also, a protocol for agreeing to the opinions of the participants should be introduced, according to which smart contracts will be updated. Such protocol could involve a majority vote and decision, a veto mechanism or other mechanisms.

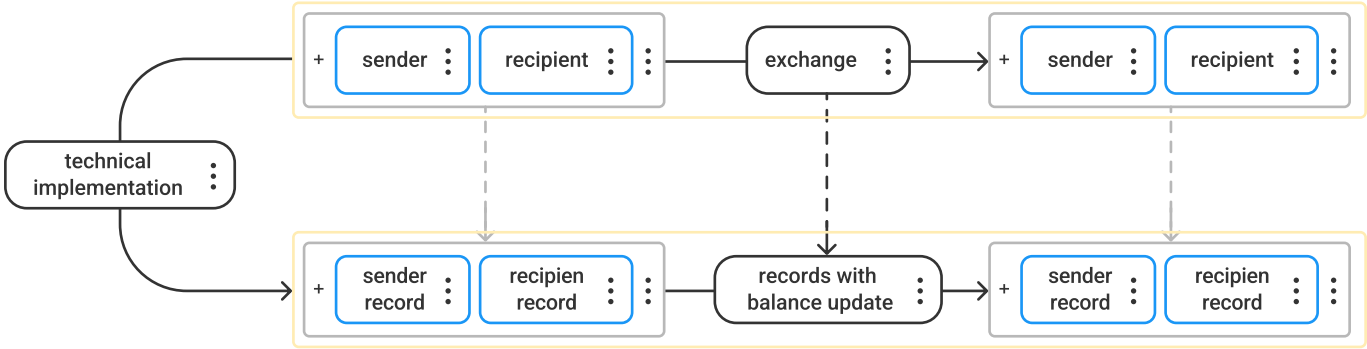
We can consider a protocol for the secure update of smart contracts, in which smart contracts can be updated only for those where an economic model is provided with proof of its consistency with the new version of smart contracts. This approach will protect the company from ill-considered changes in smart contracts without calculating an updated economic model.

Since the economic model is probabilistic, it will always be based on certain assumptions. These assumptions will be explicitly indicated in the **category-theory based proof**, and the participants of the company can separately approve all the assumptions by the approval protocol (voting, veting mechanism, etc.).

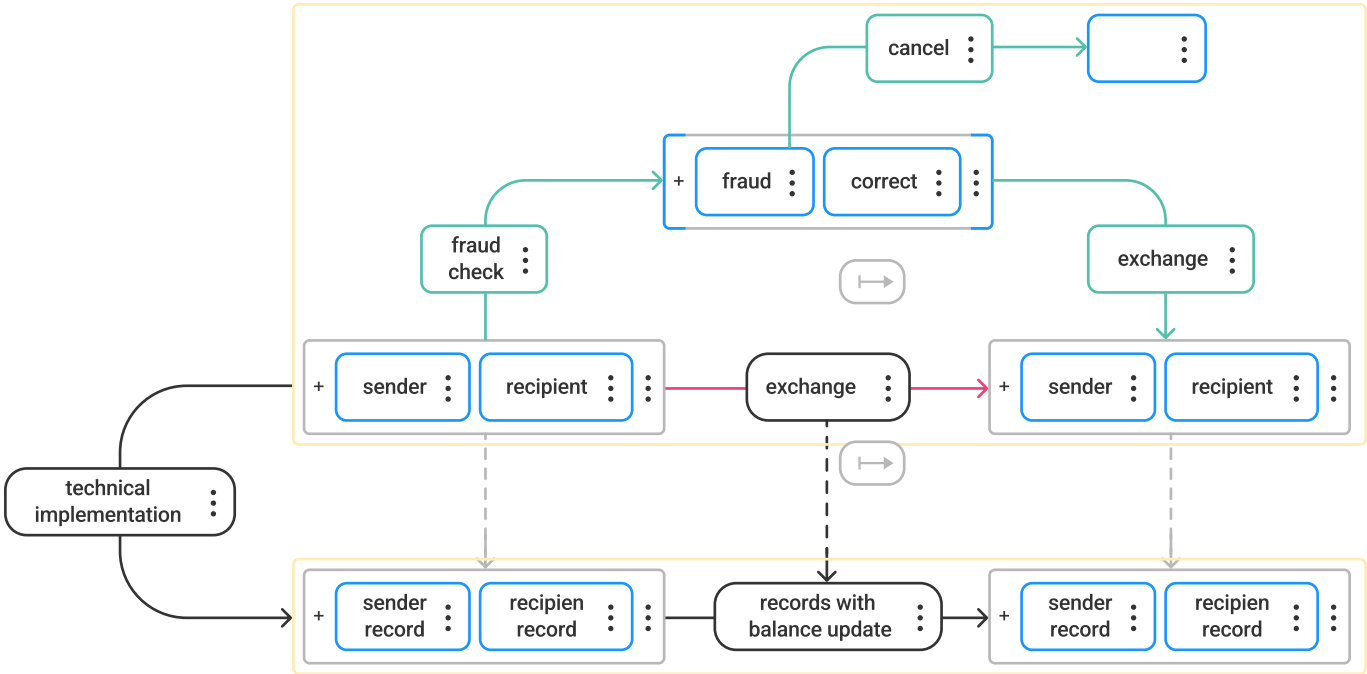
## Applications:

### Annex 1

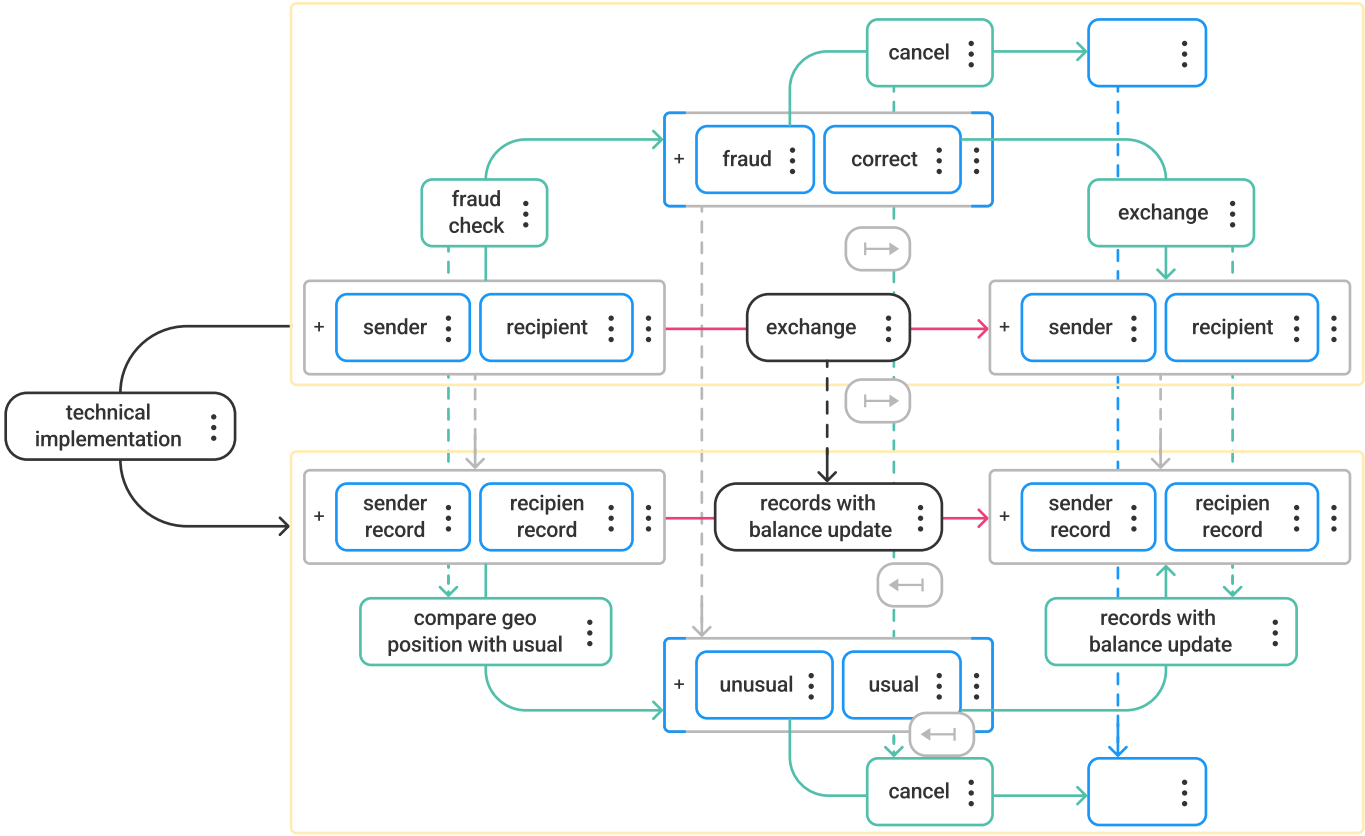
Example – implementing a fraud check in an exchange operation using a decentralized exchange.
The given example considers a decentralized exchange, in which the exchange operation is the process of updating the records of a smart contract.
The diagram of the exchange operation with the implementation will look like this:



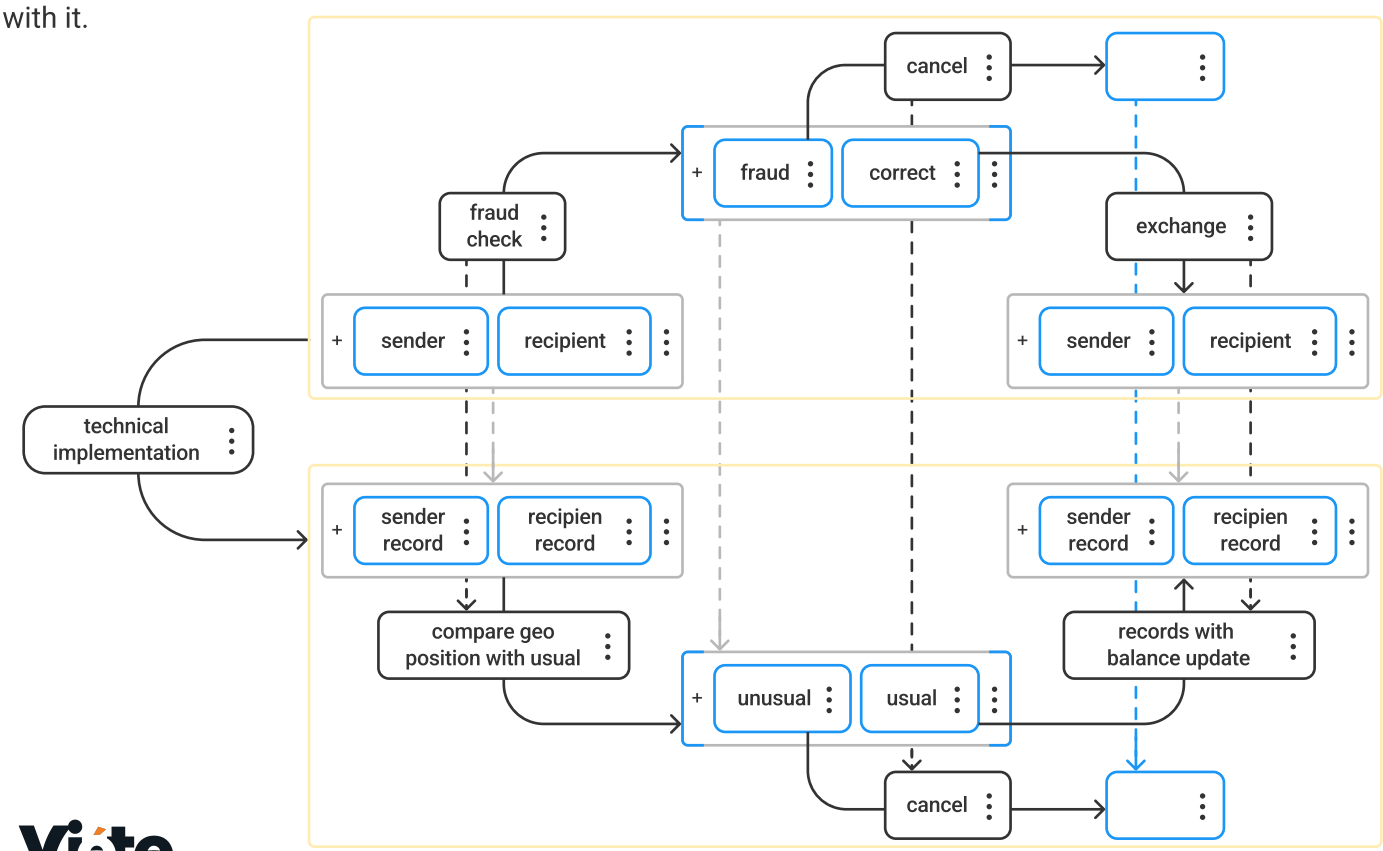

The addition of new logic with a fraud check and the removal of old logic can be described interactively:



Changes to business logic will be highlighted in color, highlighting the added, removed, and altered items. The button allows for the changes to be distributed to related models. In this case, it is the update of the technical implementation.




After the changes have been migrated, users can verify that all the changes have been introduced correctly and apply them, having received the updated business logic and the technical implementation consistent with it.



Smart-refactoring creates a process of consistently changing models that protects against errors, for example, from the fact that some of the changes in the concept will be omitted from the technical implementation.

Below is an example, when a fraud check was added in the technical implementation, but the description of the logic behind a decision to cancel an exchange operation based on the result of the verification was forgotten.