Designing a Software Solution to Increase the Conversion Rate of Transactions Through a Payment Platform

Marina Sokolova[0000-0002-7767-5220], Natalia Mamedova[0000-0002-8934-7363], Olga Staroverova[0000-0003-2605-9417], Arkadiy Urintsov[0000-0003-0273-5134]

Plekhanov Russian University of Economics, Stremyanny lane, 36, Moscow, 117997, Russia
rector@rea.ru

Abstract. The research process illustrates the results of a comprehensive analysis of the microservice architecture of a payment service provider's payment platform. Possibilities of functional development of the payment solution are revealed. A study of vulnerabilities in the functioning of the processes of conducting transactions through the payment platform was carried out. Were analyzed the specifications that set the standards for the procedures for conducting transactions. Thus, the goal of identifying the main "segments" was achieved, where the decrease in the throughput of payments is observed to the greatest extent. The existing methodological approaches to prevent payment rejection have been identified, which helps to increase the throughput of transactions. The work presents the results of designing a functional tool - "cascading". Cascading allows automatic or manual transitions through payment gateway channels. The proposed software solution increases the likelihood of making a payment, that is, it increases the conversion rate in general.

Keywords: cascading, transactions, microservice architecture, provider, payment service, payment platform

1 Introduction

In the context of the development of the digital economy, ubiquitous access to communication channels and the Internet, the flow of processed transactions significantly increases every year, as a result, the load on payment solutions is rapidly increasing [1]. This leads to a constant increase in the overall requirements for electronic payment services and, in particular, for their bandwidth. This trend not only forces payment service providers to constantly improve their product portfolios, but also raises the question of a fundamental revision of the architecture, functions and operating principles of already implemented solutions.

The fundamental fact is the desire of IT companies in the financial sector to maximize the economic efficiency of the services they provide [2]. Here, the key role is played by one of the most important performance indicators in the provider's activities - the payment conversion rate.

Despite the fact that when attempting to conduct a transaction, there may be many reasons why the payment will not actually be made, it is still possible to prevent a

Proceedings of the of the XXIII International Conference "Enterprise Engineering and Knowledge Management" (EEKM 2020), Moscow, Russia, December 8-9, 2020.

certain percentage of failures by providing the system with optimal functionality that supports multi-channel payment and payment algorithms [3]. Thus, one of the highest priority tasks in optimizing a payment solution is to increase the level of successful transactions by creating an optimal set of functional capabilities in the system. This indicator not only indicates the quality of the services provided by the providers, supporting the demand for the solution and, thereby, affirming high competitiveness, but also has a direct impact on the profit of companies and the growth of the economy of this market sector as a whole.

Now in the world market there are a large number of providers offering services for conducting electronic payments through their own payment platforms. Providers study the characteristics of each client company, operational capabilities, behavior of end users of the site and much more [4]. Based on the data received, payment solutions are customized for a specific business. So, providers provide their customers with online services of various formats, including electronic terminals and payment pages, with the help of which electronic payments are carried out in various methods.

When analyzing the functioning of payment solutions, it was found that the provider's platform has a hierarchical structure. The features of the payment platform take into account the industry focus of the company, which requires continuous operation of all elements of the system and the fastest possible processing of a large number of transactions. The functionality covered by the system includes the presence of customized interaction with a variety of acquiring banks, billing systems of providers of goods and services, card and other payment networks [5].

The coverage area of payment methods integrated into the platform plays an important role here. For each region where the provider provides its services, there is an optimal set of methods that are most popular in use. By ensuring integration with the largest number of basic payment methods for each served territory, the provider takes a strong position in its market sector and creates the potential for creating additional functionality in the system, such as multichannel payments and payments. Many of the largest providers aggregate, in addition to card payment methods, also various alternative methods [6]. In total, there are more than 3 thousand different payment methods around the world. Each method differs depending on the region of service.

Based on the above facts, it can certainly be argued that such a variety in the choice of payment methods opens up new opportunities in approaches to payment processing. The process of linear transaction execution, which is familiar to all, is limited by the rule according to which one call to the payment system occurs for one payment [7]. Of course, this approach was optimal when integrating with a small number of payment systems. However, in the current reality, a payment service provider can aggregate several hundred different payment methods. In such conditions, it is worth thinking about new algorithms that can, to one degree or another, affect the quality of the service provided.

So, the problem of low passability of successful payments solved in the framework of this work can be solved by adding new functionalities that increase the chance of successful payment. For example, such a solution can be an algorithm with a step-by-step transition through the channels of payment systems.

Thus, it was found that an important nuance in the study is the fact that such functionality is distinctive precisely for aggregating payment platforms of providers.

Aggregation here becomes a key factor that allows for the development of the system by adding tools associated with the use of several payment gateway channels for making the same payment. Without integration with a large number of payment methods in one node, the implementation of such functionality would be significantly less efficient.

2 Materials and Methods

The work is devoted to the design of a software solution as part of the payment platform of a payment service provider to increase the level of transaction conversion. A comprehensive analysis of the architecture of the payment platform of the payment service provider was carried out and the possibilities of the functional development of this payment solution were identified. It has been established that in order to maintain a high level of demand, a payment solution must have optimal structural components that allow for its potential development [8]. As a result, an easily scalable IT infrastructure must be built, and a multifunctional architecture of the payment platform must be designed.

In the process of performing the work, it was determined that the provider's system, in the most frequent cases, has a microservice architecture that supports the operation and interaction of several separate components responsible for specific functional tasks. Thanks to the microservice architecture, the platform achieves isolation of component failures: the main elements can continue to work efficiently, even if a single module fails. Figure 1 shows a typical architecture of a payment solution that displays the interaction of the services of a payment platform.

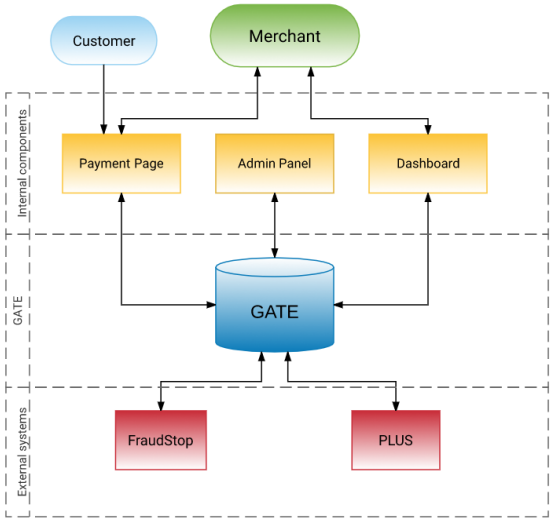

Fig. 1. Microservice architecture of the payment platform

It was also revealed that the central element of the architecture of the processing center is the core, through which interactions with all major services and agents take place. The core is the main component that implements the logic of conducting

transactions on available means of payment in accordance with the settings passed in the request for payment or set in the control component. This component is directly responsible for creating the "payment" and "operation" entities, generating and transmitting requests for a transaction, receiving and processing responses from external payment systems. So, the most important aspect here is the possibility of introducing various types of additional functionality into the core of the payment platform, imposed on the payment procedure (Fig. 2).

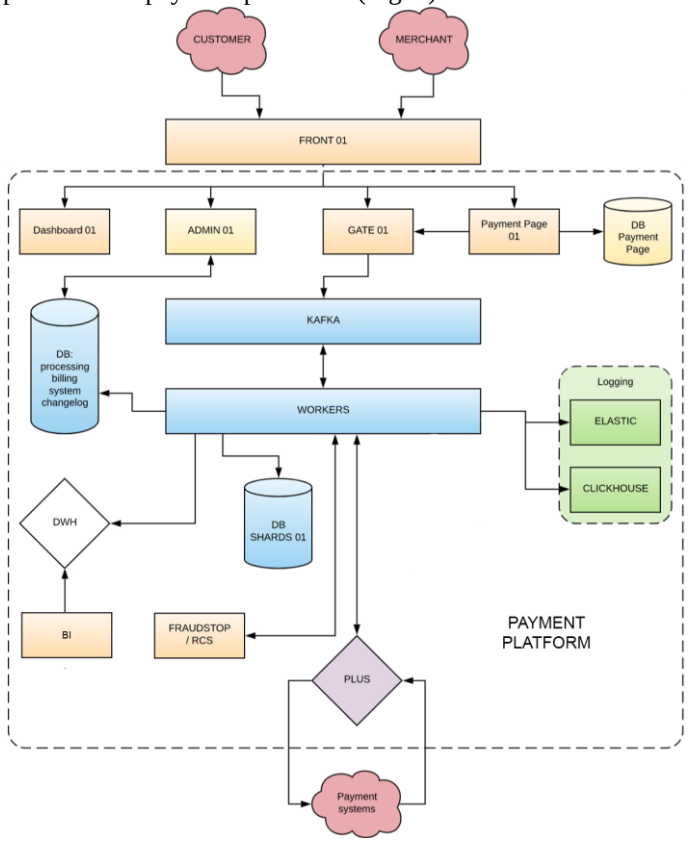

Fig. 2. Microservice architecture of the payment platform with display of technical components

Thus, it has been established that the considered microservice architecture is capable of providing the system with all the necessary functionality, which allows us to provide a company with a high-quality service. In addition to supporting optimal protocol solutions, ensuring high performance in the operation of the system and guaranteeing the effective operation of the company, the architecture of the platform allows the implementation of various narrowly focused functionality that contributes to maintaining a high economic level of the payment solution.

A study of vulnerabilities in the functioning of the processes of conducting transactions through the payment platform, as well as specifications that establish the norms of procedures for conducting transactions, was carried out in order to identify

the main “segments” where the decrease in the throughput of payments is most observed.

In the course of the work, it was found that in addition to the need to ensure the security of the payment system and increase the speed of transactions, it is equally important to maintain a high throughput of successful payments. Thus, it was revealed that the main indicator of the efficiency of the provider's payment platform is the level of payment conversion [9, 10].

It has been determined that there are many reasons why the conversion value is often not high enough, many of which are practically impossible to resolve, while others can be partially eliminated.

So, the main reasons are highlighted:

- Buyer's mistakes.
- The buyer shows suspicious activity and is blocked by anti-fraud systems.
- The transaction was declined by the issuing bank or acquiring bank.
- Errors 3D-S authorization.
- The transaction was rejected on the side of the payment system.
- The transaction is automatically rejected (autodecline) by the provider's system because the payment system has exceeded the waiting time for a response.

Of course, the higher the conversion rate provided by the system, the higher the quality level of the services provided by the provider and the greater the amount of profit in his company. As a result of the study, it was determined that conversion problems often arise due to lack of flexibility and insufficient technical capabilities of the payment gateway. Most of the factors leading to a decrease in the conversion rate cannot be completely eliminated, however, some of them are still amenable to regulation, thanks to the creation of special settings and the development of additional functionality in the payment platform.

The existing methodological approaches to prevent rejection of payments are identified, which contribute to an increase in the level of throughput of transactions.

It has been established that in order to maintain the conversion of payments at a high level, it is necessary to ensure the throughput of successful transactions, which can be achieved by the functionality of the platform. Since there are many reasons for transaction failures, providers are actively developing methods to prevent them. So, the main reasons leading to a decrease in the conversion rate were identified, and the most common solutions to this problem were analyzed, including:

- UX and UI approach to preventing incorrect data entry and other user errors when making a payment request.
- Regulatory measures to ease restrictive conditions for blocking payments by anti-fraud systems.
- Regulatory measures to prevent denials when conducting transactions through the 3DS-scheme.

Also, it was revealed that often the main reasons for the decrease in conversion are the unstable operation of the servers of payment systems or the refusal to conduct a transaction on the side of payment systems. In search of a solution to this problem, it was found that having a standard tool for creating and operating transactions and the presence of a large number of integrations with the payment system, the provider can expand the capabilities of its payment platform and add algorithms that significantly

increase the level of conversion rate and the service provided by the company as a whole. The design of one of these tools served as the basis for the work.
The theoretical aspects were analyzed, establishing some fundamental basis [11-12], on the basis of which the concept of functional tools for increasing the level of conversion of payments by preventing complete rejection of the transaction due to refusal or other problems on the side of the payment system became possible.
Also, the main entities formed during the transaction were identified. It has been established that the components “payment” and “operation” carry different functional loads, which makes it possible to form the concept that rejection of a transaction does not mean rejection of the payment as a whole. In this case, it is important to approve the following rule: the failure of the payment system to carry out the first operation does not limit the possibility of forming and attempting to make a payment through the second and further operations, which, of course, can become a fundamental factor in designing a solution to increase the conversion of payments.

3 Results Of the Study

The project solution for the development of functional tools to increase the level of conversion of transactions consists in the implementation of a step-by-step transition through the channels of payment systems and transmission of requests for payment to them. The following theses were formulated to substantiate this decision.

1. It has been determined that one of the main features of the provider's payment platform is the aggregation functionality. It allows you to integrate in one solution a large number of different payment systems that support a variety of payment methods, among which there are both card payment methods and alternative methods.
2. It was found that the provider's payment platform has a microservice architecture. It contains the central component Gate, which is directly responsible for creating the “payment” and “operation” entities, generating and transmitting requests for a transaction, receiving and processing responses from external payment systems. The functionality of the Gate component can be extended with additional tools.
3. It was revealed that the main indicator of the efficiency of the provider's payment platform is the level of payment conversion. Since there are many reasons for transaction failures, providers are actively developing methods to prevent them. One of the most significant and unresolved reasons is the unstable operation of the servers of payment systems or the refusal to conduct a transaction on the side of the payment system.
4. A functional solution was proposed that contributes to an increase in the percentage of successful payments, despite the possibility of frequent rejection of transactions on the side of payment systems. Thus, based on the obtained theoretical knowledge, for a practical study of the issue, the goal was set - to design a tool for multi-stage payment execution – “cascading”, where each stage is a separate request to perform an operation through a certain provider.

This section details all the important points related to the design solution.

When concluding an agreement for the provision of payment services with a specific merchant, several technical entities are created that are used in the process of making payments. The “merchant” entity contains basic legal and technical information about the merchant and its system. One merchant can have several projects (entity “project”), each of which is associated with a specific resource of the merchant - an online store, a website with services provided, etc.

For each project, there are many settings that affect the processes within the payment platform. So, one of the settings set in Backoffice for a project is to specify certain payment methods and systems with which the merchant has concluded agreements for cooperation and registration in these systems. Specifying the payment system means that making a payment for the current merchant's project is possible through this channel. It is also permissible to configure the project in such a way that the lists of the payment system will be different for the payment types “payment” and “payout”.

In addition to indicating the very list of acceptable payment systems, their percentage breakdown is also indicated. For example, if, in accordance with the agreement, the commission for making a payment through a certain payment system is significantly less than through another payment system, it is more profitable for the merchant to request the installation of a higher percentage in the settings of the first system than the second. Before making a payment within the platform, the Gate component, based on the percentage specified in the project, performs the process of calculating the payment system - the channel where it is planned to send a request for payment or payment.

In accordance with the logic of the cascading operation, several requests can be sent to different payment systems within one payment. For this, Gate needs to calculate and form in the correct order a list of payment systems that could process the payment requested by the user. When calculating, other project settings are also taken into account - for example, for some payment systems, only certain currencies are indicated, for others, limits on the permissible amounts of payments can be set. As it was specified earlier, several operations can exist within the payment entity. As part of a cascading transaction, the transactions in the payment will be of the same type, for example, type “sale” or “payout”. It is important to note a nuance that from the point of view of the merchant, the completion of the operation does not mean the completion of the payment as a whole, therefore, the status of the operation and the status of the payment within the payment platform are different parameters.

Thus, the first process in the functioning of cascading is the calculation of a list of payment systems, requests to which will be generated in separate operations. Further, after successfully calculating the array of payment systems, the platform, and more specifically the Gate service, generates and sends a request to the first payment system from the list. In accordance with the API of the payment system, the request is composed in a specific format, more often in JSON, and sent to a specific entry point. Further logic of the instrument operation will differ depending on the type of cascading.

Cascading of the first type is the formation of an algorithm for the functioning of cascading with automatic step-by-step.

After a request has been sent to the first payment system, Gate expects a callback in accordance with an asynchronous interaction scheme. If the received response contains the status “success” with a code indicating a successful transaction, the

payment status will change to “success” within the platform, after which Gate will send a callback with payment information to the merchant, and the user will be redirected to the page about successful payment. So, the payment will be completed.
However, if the “decline” status is indicated in the response from the PS, Gate will have to verify the received code. Depending on the reason for the refusal, determined by the code, the system will either complete the payment with a refusal or proceed to the next step. In the first case, a callback will be immediately sent to the merchant with information that the transaction was declined, and the user will be redirected to the refusal page. In the second case, the cascading functionality will work, and when moving to the second step, Gate will create a new operation in the database as part of the previous payment and at the same time send a request to the next payment system from the list. The process will be repeated until the moment when:

- in response from the payment system, a status will be received about the successful completion of the operation, which will lead to the change in the payment status also to “success”;
- the response from the payment system will receive the “decline” status with a response code that does not allow further cascading;
- the last step will be taken, and there will be no more channels in the list of the payment system to which the request has not yet been sent.

The algorithm can become more complicated in the case of making a payment through a bank card. Before each call to the payment system, it is necessary to redirect the user to the ACS page to enter the 3DS code, if the payment system requests such a validation. Of course, from the point of view of communicating with the user on the payment page, it is necessary to display an instruction explaining that multiple requests for the 3DS code are possible during payment.
The payment process in this case will proceed as follows. The cascading algorithm contains several steps. The user is redirected to the payment page and fills out the payment form (enters card data), then he confirms the payment. A request for payment is received from PP to Gate, payment is created payment_id = 1, a list of payment systems is calculated. The cascading process begins, and a request is made to the payment system according to the first step (3DS MA means that the introduction of the 3DS code is required for the payment system). The system receives acs_url from payment systems, and the user is redirected to the link, after which he receives a code via SMS and enters it. Gate sends a request for payment and receives a response from the payment system: decline and code / message, after which Gate checks if there are still pending steps in the cascading. If there are still steps left, the cascading continues. The user is redirected to the issuer's page again, where he enters a new SMS code. The process is repeated.
Cascading of the second type is the formation of an algorithm for the functioning of cascading with a manual step-by-step mechanism.
The cascading operation algorithm with a manual step-by-step mechanism is similar to the cascading logic of the first type. A distinctive feature is the fact that, in the first case, the payment system sends a response with the status and code of the transaction, and in the second case, the payment system may encounter problems in the system on its side and for this reason either send a response after a long time, or it won't send it at all.

Thus, after a request has been sent to the first payment system, Gate expects a callback in accordance with an asynchronous interaction scheme. The payment and the first operation at this time are in the waiting status. Further, in accordance with the logic of functioning of the second type of cascading, the user must independently initiate the transition to the next step. To do this, a special button is displayed on the payment form and a corresponding instruction describing the possible actions of the user and the risks that he assumes.

Pressing the button will start the transition to the second step of the cascading, while the status of the operation on the first request will also remain unknown. The process can be repeated until there is no one among the payment systems calculated in the list that has not yet received a request. Here, in the process of cascading, there is a risk of payment doubling. Let's say the user goes to the second step of the cascading, and the platform sends a request to another payment system. The response will receive "success", the payment status will also be changed to "success", the system will redirect the user to the page informing about successful payment, and at the same time pass the callback to the merchant. It would seem that the payment process is complete. However, after some time, an answer may come from the payment system about the successful completion of the transaction from the first step. This will mean that the money was transferred twice. In this case, the payment platform must notify the merchant about an existing successful second transaction for this user and either independently create return transactions, or wait for a return request from the merchant. Thus, the use of manual cascading in the payment platform is impossible without the functionality of returns.

Since cascading provides for the possibility of successfully carrying out several operations within one payment, for each of which a callback will be sent to the merchant, the callbacks must contain all the necessary data, on the basis of which the merchant can unambiguously determine:

- the fact of multiple payments;
- channels through which the payment was made.

So, it should be borne in mind that some parameters of callbacks are not mandatory for transferring to merchants. But within the framework of using the cascading functionality, it is necessary to transfer complete information on the payment and on the operation.

4 Findings

In order to increase the conversion rate of payment transactions, a tool was designed that, in the event of an unsuccessful payment transaction, allows the use of several consecutive attempts to effect a payment through various available channels.

The software solution significantly increases the share of successfully completed payments, since it provides the ability to use backup channels for outgoing and incoming payments, and also contributes to an increase in the total limits on transactions, since it allows you to automatically use alternative available channels when the limits are refused.

The implementation of the cascading mechanism designed within the framework of the work is based both on automatic switching between existing payment channels

and on manual switching during processing of a specific transaction in cases when the system receives response codes specified in the settings from external providers. This allows you to flexibly customize the conditions of reaction to one or another behavior of external payment systems. Combined with customized filters for each project, cascading allows for significantly more flexibility in using various combinations of available payment settings to maximize the reduction in the share of unsuccessful payments.

References

1. Digital economy, digital money and digital banking, Series: Economics and Organization, 6, No 2, 153-160 (2009)
2. Semenyuta, O.G., Repkina, I.V.: On the question of the economic essence of the concept and contentment "payment system". Financial Research, 3 (44) (2014)
3. Karavai Kund, B.: Electronic payment systems and their possible way of development. The World, 12 (2017)
4. Hasselbring, W.: Microservices for scalability. ICPE 2016 - Proceedings of the 7th ACM/SPEC International Conference on Performance Engineering, 133-134 (2016)
5. The Payment Card Industry Data Security Standard - PCI DSS Applicability in an EMV Environment / Security Standards Council. A Guidance Document. Ver. 1 (2011)
6. Zhuravleva, T.A.: The National Payment System in Russia: Prerequisites of Emergence and Current Status. Discussion, 2 (65), 54-60 (2016)
7. De Brock, B.: Declarative specifications of complex transactions with an application to cascading deletes. Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), 1773, 150-166 (2000)
8. Limon, X., Guerra-Hernandez, A., Sanchez-Garcia, A.J., Perez Arriaga, J.C.: SagaMAS: A software framework for distributed transactions in the microservice architecture. Proceedings - 2018 6th International Conference in Software Engineering Research and Innovation, CONISOFT 2018, 8645853, 50-58 (2019)
9. Miyake, S., Mashita, K., Yamada, R., Tsumura, T.: Eliminating Cascading Stall on Hardware Transactional Memory. Proceedings - 2015 3rd International Symposium on Computing and Networking, CANDAR 2015, 7424703, 147-153 (2016)
10. Wang, L., Li, Q., Zong, X.: Distributed optimization for energy transactions and production of multiple microgrids under uncertainty. Proceedings - 2019 Chinese Automation Congress, CAC 2019, 8996866, 1334-1338 (2019)
11. Wakamori, N.: Why do shoppers use cash? Evidence from shopping diary data. Journal of Money, Credit and Banking, 49(1), 115-169 (2017)
12. Liu, Y., Liang, Y., Hu, Y.: A fair payment protocol for cascading P2P transaction. Journal of Networks, 6(9), 1313-1320 (2011)