PROCESSING METHOD AND APPARATUS FOR PAYMENT SERVICE, STORAGE MEDIUM, AND COMPUTER SYSTEM

TECHNICAL FIELD

[0001] One or more embodiments of this specification relate to the field of computer technologies, and in particular, to a processing method and apparatus for a payment service, a storage medium, and a computer system.

BACKGROUND

[0002] Electronic payments allow individuals and enterprises to quickly and securely complete transactions through the Internet or other digital communication technologies. Specific forms include online bank transfers, electronic wallet payments, and the like. Currently, there are payment suppliers providing cross-border electronic payment services, greatly promoting the development of global commercial activities.

[0003] In a merchant-presented mode scenario of cross-border consumption, users scan collection two-dimensional codes of overseas merchants by using electronic wallets in home countries, to make payments. However, currently, a related payment link cannot meet a higher requirement in actual applications. Therefore, the embodiments of this specification provide a payment link improvement solution for a cross-border merchant-presented mode scenario, to better meet an actual application requirement, for example, reduce a payment time, so as to improve user experience.

SUMMARY

[0004] Embodiments of this specification describe a processing method and apparatus for a payment service, to effectively reduce a payment time in a cross-border merchant-presented mode scenario by optimizing a technical link, so as to greatly improve payment experience of a user.

[0005] According to a first aspect, a processing method for a payment service that involves at least an acquiring institution and a third-party platform in a first region and a wallet serving end

in a second region is provided. The method is applied to the third-party platform and includes: generating a third-party order number based on a first order placement request received from an upstream platform, where the first order placement request includes information about a target payment service, and the target payment service is initiated by a user by scanning a collection two-dimensional code by using a wallet app corresponding to the wallet serving end; sending a second order placement request to the wallet serving end, and without waiting for return of an order placement success result corresponding to the second order placement request, directly sending a target notification indicating the order placement success result to the upstream platform; receiving an order number replacement request sent by the wallet serving end, where the order number replacement request carries an acquiring order number generated by the acquiring institution for the target payment service, and the acquiring order number is provided by the acquiring institution to the wallet app in response to the target notification, and is sent by a third-party SDK embedded in the wallet app to the wallet serving end; and performing target processing on the order number replacement request, where the target processing includes: providing the third-party order number corresponding to the acquiring order number to the wallet serving end if the order placement success result is returned, so that the wallet serving end provides cashier interface information to the wallet app based on the third-party order number, to complete a target payment.

[0006] According to a second aspect, a computer system is provided and includes a memory, a processor, and a computer program that is stored in the memory and that is capable of running on the processor. The processor is integrated into a third-party platform, and the third-party platform and an acquiring institution are located in a first region. When the processor executes the program, the following steps are implemented: generating a third-party order number based on a first order placement request received from an upstream platform, where the first order placement request includes information about a target payment service, the target payment service is initiated by a user by scanning a collection two-dimensional code by using a wallet app corresponding to a wallet serving end, and the wallet serving end is located in a second region; sending a second order placement request to the wallet serving end, and without waiting for return of an order placement success result corresponding to the second order placement request, directly sending a target notification indicating the order placement success result to the upstream platform; receiving an order number replacement request sent by the wallet serving end, where the order number replacement request carries an acquiring order number generated by the acquiring institution for

the target payment service, and the acquiring order number is provided by the acquiring institution to the wallet app in response to the target notification, and is sent by a third-party SDK embedded in the wallet app to the wallet serving end; and performing target processing on the order number replacement request, where the target processing includes: providing the third-party order number corresponding to the acquiring order number to the wallet serving end if the order placement success result is returned, so that the wallet serving end provides cashier interface information to the wallet app based on the third-party order number, to complete a target payment.

[0007] According to a third aspect, a processing apparatus for a payment service is provided. The payment service involves at least an acquiring institution and a third-party platform in a first region and a wallet serving end in a second region. The apparatus is integrated into the third-party platform and includes: a third-party order number generation module, configured to generate a third-party order number based on a first order placement request received from an upstream platform, where the first order placement request includes information about a target payment service, and the target payment service is initiated by a user by scanning a collection two-dimensional code by using a wallet app corresponding to the wallet serving end; an asynchronous processing module, configured to send a second order placement request to the wallet serving end, and without waiting for return of an order placement success result corresponding to the second order placement request, directly send a target notification indicating the order placement success result to the upstream platform; a replacement request receiving module, configured to receive an order number replacement request sent by the wallet serving end, where the order number replacement request carries an acquiring order number generated by the acquiring institution for the target payment service, and the acquiring order number is provided by the acquiring institution to the wallet app in response to the target notification, and is sent by a third-party SDK embedded in the wallet app to the wallet serving end; and a target processing module, configured to perform target processing on the order number replacement request, where the target processing includes: providing the third-party order number corresponding to the acquiring order number to the wallet serving end if the order placement success result is returned, so that the wallet serving end provides cashier interface information to the wallet app based on the third-party order number, to complete a target payment.

[0008] According to a fourth aspect, a computer-readable nonvolatile storage medium is provided. The computer-readable nonvolatile storage medium stores a computer program, and

when the computer program is executed on a computer, the computer is enabled to perform the method according to the first aspect.

[0009] In conclusion, according to the method and apparatus provided in the embodiments of this specification, asynchronous processing is performed on order placement with a wallet, to effectively reduce a payment time in a cross-regional merchant-presented mode service scenario. Further, an asynchronous retry for order number replacement is introduced, to resolve a possible problem that procedure asynchronization affects a payment success rate, and achieve a win-win situation for a payment time and a service success rate. Further, an automated retry parameter configuration manner is formed in the asynchronous retry, to ensure 100% success of a first retry in a non-system exception case. In addition, system stability is ensured by using a plurality of retry parameters designed for a system exception case.

BRIEF DESCRIPTION OF DRAWINGS

[0010] To describe the technical solutions in the embodiments of the present invention more clearly, the following briefly describes the accompanying drawings required for describing the embodiments. Clearly, the accompanying drawings in the following description show merely some embodiments of the present invention, and a person of ordinary skill in the art can still derive other drawings from these accompanying drawings without creative efforts.

[0011] FIG. 1 is a schematic diagram of multi-party interaction in a cross-regional payment service;

[0012] FIG. 2 is a schematic diagram of an interface change in a wallet app when a user uses a cross-regional payment service;

[0013] FIG. 3 is a schematic diagram of procedure steps of a processing method for a payment service applied to a third-party platform according to an embodiment of this specification;

[0014] FIG. 4 is a schematic diagram of multi-party interaction after asynchronous improvement is made based on order placement with a wallet in FIG. 1 according to an example;

[0015] FIG. 5 is a comparison diagram of timing before and after asynchronous order placement improvement to a payment link is made according to an embodiment of this specification;

[0016] FIG. 6 is a schematic structural diagram of a processing apparatus for a payment service

integrated into a third-party platform according to an embodiment of this specification; and

[0017] FIG. 7 is a schematic structural diagram of a computer system according to an embodiment of this specification.

DESCRIPTION OF EMBODIMENTS

[0018] The solutions provided in this specification are described below with reference to the accompanying drawings.

[0019] As described above, currently, a conventional payment link for a cross-border merchant-presented mode scenario cannot meet a higher requirement in actual applications. Specifically, the conventional payment link is first described below with reference to FIG. 1 and FIG. 2, and then a key problem discovered by the applicant by performing breakdown analysis on the link is described.

[0020] As shown in FIG. 1, a plurality of terminals involved in a cross-border payment link (or referred to as a cross-regional payment link) include a merchant serving end, a wallet serving end, a wallet app, a third-party platform, an acquiring institution (or referred to as an acquiring platform), and a regulatory platform.

[0021] The merchant serving end is located in a first region, and the wallet serving end is located in a second region different from the first region. Based on this, when a user enters the first region (for example, goes on a business trip or travels) from the second region, and executes a transaction with a merchant in the first region by using the wallet app corresponding to the wallet serving end, a cross-regional payment service scenario is generated. Further, the embodiments of this specification focus on a merchant-presented mode scenario in a cross-regional payment. A merchant-presented mode means that a user scans a collection two-dimensional code of a merchant. It should be understood that a consumer-presented mode is opposite to the merchant-presented mode, and means that a merchant scans a payment two-dimensional code of a user. For brevity of description, in this specification, the merchant-presented mode scenario in the cross-regional payment is referred to as a cross-regional merchant-presented mode scenario. In addition, it should be noted that the region in this specification can generally refer to a country; that a serving end or a platform is located in a region in this specification mainly means that a physical server that corresponds to the serving end or the platform and that is used for software running and data

storage is located in the region; and "first" and "second" in "first region" and "second region", and similar terms in other parts of this specification are intended to distinguish between similar things of a same type and do not imply a sequence or constitute other limitations.

[0022] The third-party platform is located in the first region, and can provide users with a capability of making cross-regional payments. With reference to FIG. 1, it can be seen that the third-party platform is a bridge between the merchant serving end and the wallet serving end. When performing step S108, the third-party platform converts an amount in a currency corresponding to the first region in a merchant order into a second amount in a currency corresponding to the second region based on an exchange rate, and then provides the second amount to the wallet serving end for creating a wallet payment order. It should be understood that the third platform may actually further involve compliance constraints on cross-regional payment services, and the like. However, this is not a focus of this solution, and details are not described.

[0023] The acquiring institution is located in the first region, plays a key role in a payment system, and is used to ensure security, convenience, efficiency, and the like of transactions. Almost every country or region has its own acquiring institution.

[0024] The regulatory platform is located in the first region, has functions including regulating the electronic payment market, ensuring compliance of payment transactions, and the like, and sometimes performs a clearing function. There may alternatively be no regulatory platform in the payment link, or the acquiring institution and the regulatory platform can be integrated. In this case, steps S106 and S107 can be replaced as a whole with the following: The acquiring institution places an order with the third-party platform. Similarly, steps S111 and S112 can be replaced as a whole with the following: The third-party platform sends an order placement success notification to the acquiring institution.

[0025] The terminals participating in interaction in FIG. 1 are preliminarily described above. The following interaction steps shown in FIG. 1 are sequentially described below.

[0026] Step S101: A user scans, by using a wallet app, a collection two-dimensional code provided by a merchant. It should be understood that for clarity of description, a payment service initiated with execution of this step is referred to as a target payment service.

[0027] For example, the collection two-dimensional code can be an aggregation code. It should be understood that the aggregation code is also referred to as "one code for a plurality of payments" or "one code pass", and is a two-dimensional code technology. The aggregation code integrates

payment functions of a plurality of payment platforms into a single two-dimensional code, and is currently a relatively popular payment two-dimensional code technology.

[0028] Usually, the user can perform a code scanning operation by using the wallet app installed in a mobile terminal such as a mobile phone or a wearable device of the user. For example, the user can tap a Scan icon in FIG. 2 to use a code scanning function. Further, FIG. 2 further shows a display interface of the mobile terminal during scanning of a two-dimensional code.

[0029] Step S102: The wallet app displays a corresponding store page by parsing the scanned collection two-dimensional code.

[0030] It should be understood that the store page includes some information and function options related to the merchant. The information and function can include: (1) a name or a trademark of the merchant; (2) a specific payment amount or a payment amount input box; (3) product information; (4) discount information provided by the merchant; and (5) a confirm button.

[0031] For example, refer to a store page that is displayed after scanning succeeds and that is shown in FIG. 2.

[0032] Step S103: The user sends a confirmation operation based on the store page.

[0033] In an embodiment, the store page includes the specific payment amount and the product information. In this case, the user can directly tap the confirm button. In another embodiment, the store page includes the payment amount input box. In this case, the user can tap to confirm after entering the amount, for example, can tap a "confirm to pay" icon on the store page shown in FIG. 2.

[0034] It should be understood that the merchant belongs to the first region, and in this specification, an official currency in the first region is a first currency. Correspondingly, the payment amount in the first currency entered by the user is referred to as a first amount.

[0035] Step S104: The wallet app initiates a third order placement request to the merchant serving end based on the confirmation operation sent by the user, where the third order placement request includes third information about the target payment service. It should be noted that the first order placement request and the second order placement request mentioned in the summary are used to refer to other order placement requests. Therefore, to ensure consistency and avoid confusion throughout the specification, an order placement request initiated in this step is referred to as the third order placement request, and information included in the third order placement request is correspondingly referred to as the third information.

[0036] For example, the third information includes the first currency, the to-be-paid first amount, the product information, a name of the wallet app, an identity (for example, a mobile phone number, an email address, or a document number) of a login user in the wallet app, and the like.

[0037] Step S105: After creating a corresponding merchant order based on the received third order placement request, the merchant serving end initiates a fourth order placement request to the acquiring institution.

[0038] For example, the merchant serving end first constructs the merchant order by using the third information obtained through parsing from the third order placement request and a merchant-side order number generated for the target payment service, then includes information in the merchant order in the fourth order placement request, and sends the fourth order placement request to the acquiring institution.

[0039] Step S106: After creating a corresponding acquiring order based on the received fourth order placement request, the acquiring institution initiates a fifth order placement request to the regulatory platform.

[0040] For example, the acquiring institution first constructs the acquiring order by using the fourth information obtained through parsing from the fourth order placement request and an acquiring order number generated for the target payment service, then includes information in the acquiring order in the fifth order placement request, and provides the fifth order placement request to the regulatory platform.

[0041] Step S107: The regulatory platform sends a first order placement request to the third-party platform based on the received fifth order placement request.

[0042] It should be understood that in this step, the regulatory platform usually performs some compliance check and the like on the fifth order placement request, and if the check succeeds, sends the first order placement request to the third-party platform. Otherwise, a current procedure is suspended. For example, the first order placement request includes fifth information obtained through parsing from the fifth order placement request.

[0043] Step S108: After generating a second order placement request based on the received first order placement request, the third-party platform sends the second order placement request to the wallet serving end.

[0044] Typically, the third-party platform first creates a third-party order by using first

information obtained through parsing from the first order placement request and a third-party order number generated for the target payment service.

[0045] In an embodiment, the first information can include the product information corresponding to the target payment service, the to-be-paid first amount in the first currency, and the identity of the login user in the wallet app.

[0046] In an embodiment, the creation of the third-party order can include: converting the first amount in the first currency included in the first information into a second amount in a second currency based on a real-time exchange rate between official currencies in the first region and the second region, to include, in the third-party order, an identifier of the first currency, the first amount, an identifier of the second currency, the second amount, the third-party order number generated for the target payment service, and the product information and the identifier of the login user that are included in the first information.

[0047] Further, the third-party platform can generate the second order placement request based on information about the third-party order. The second order placement request can include the second currency, the second amount, the product information, the identifier of the login user, and the third-party order number. It should be understood that the to-be-paid amount included in the second order placement request is converted into an amount in the second currency. This is because a case in which the wallet serving end is usually designed to support only processing of a payment request in a local currency is considered.

[0048] Step S109: The wallet serving end creates a corresponding wallet payment order based on the received second order placement request.

[0049] For example, the wallet serving end can perform validity check on the identifier of the login user in the second order placement request, and if validity check on a wallet identity succeeds, determine, with reference to some payment discount information, the second amount, and the like that are provided on a wallet side, a third amount that actually needs to be paid, to create a wallet-side payment order that includes the third amount, the product information, a wallet-side order number generated for the target payment service, and the like.

[0050] Step S110: The wallet serving end returns an order placement response corresponding to the second order placement request to the third-party platform.

[0051] For example, when indicating that order placement succeeds, the order placement response includes information in the wallet-side payment order, for example, the third amount and

the wallet-side order number, and can further include the third-party order number, which is used to help the third-party platform locate the target payment service.

[0052] Step S111: The third-party platform sends a first order placement success notification to the regulatory platform.

[0053] For example, the third-party platform stores data obtained through parsing from the order placement response in a database, and further, if the data obtained through parsing indicates that the order placement on the wallet side succeeds, sends the first order placement success notification to the regulatory platform.

[0054] Step S112: The regulatory platform sends a second order placement success notification to the acquiring institution.

[0055] For example, the regulatory platform retrieves the corresponding merchant-side order number based on the third-party order number carried in the first order placement success notification, then includes the merchant-side order number in the second order placement success notification, and provides the second order placement success notification to the acquiring institution.

[0056] Step S113: The acquiring institution returns the acquiring order number to the merchant serving end.

[0057] For example, the acquiring institution retrieves the corresponding acquiring order number based on the merchant-side order number in the second order placement success notification, and then returns the acquiring order number to the merchant serving end.

[0058] Step S114: The merchant serving end returns the acquiring order number to the wallet app.

[0059] It should be understood that step S114 and a service link corresponding to the previous steps are mainly used to create a cross-regional payment order at the wallet serving end. Only after the wallet app receives indication information indicating that the order is successfully created, for example, the acquiring order number in this step, the wallet app can further initiate a subsequent payment operation, to implement cash collection and deduction on the wallet side.

[0060] In addition, to facilitate understanding of subsequent steps, Table 1 below shows an order number that is held by each serving end or platform and that can be used to locally locate the target payment service by performing step S114 and a previous order creation link, and an order number that cannot be used to locate the target payment service even if the order number is

obtained is denoted as a symbol × in Table 1 below:

Table 1

	Merchant serving end	Acquiring institution	Regulatory platform	Third-party platform	Wallet serving end
Merchant order number	√	√	√	×	×
Acquiring order number	×	√	√	√	×
Third-party order number	×	×	√	√	√
Wallet order number	×	×	√	√	√

[0061] Step S115: A third-party SDK embedded in the wallet app triggers sending of an interception request to the wallet serving end, where the interception request carries the acquiring order number. It can be understood that both the third-party SDK and the third-party platform belong to a third-party.

[0062] Step S116: The wallet serving end sends an order number replacement request to the third-party platform, where the order number replacement request carries the acquiring order number in the interception request.

[0063] It should be noted that the wallet serving end cannot identify the acquiring order number because the wallet serving end has no way to establish a mapping relationship between the acquiring order number and the wallet-side order number before this step. This is because after the wallet serving end receives the second order placement request in step S108, the second information obtained through parsing does not include the acquiring order number. Therefore, the wallet serving end needs to obtain the third-party order number that can be identified by the wallet serving end from the third-party platform through replacement by using the acquiring order number. The third-party platform can monitor a subsequent process of the target payment service by accepting the order number replacement request, to facilitate service alignment with the wallet side.

[0064] Step S117: The third-party platform finds the corresponding third-party order number based on the acquiring order number.

[0065] Step S118: The third-party platform returns the found third-party order number to the wallet serving end.

[0066] Step S119: The wallet serving end locates the corresponding wallet-side payment order based on the third-party order number, and processes the wallet-side payment order, including generating a cashier address for the wallet-side payment order.

[0067] Step S120: The wallet serving end provides the generated cashier address to the wallet app.

[0068] Step S121: The wallet serving end renders a cashier interface based on the cashier address.

[0069] For example, the third amount that should be paid, the product information, a payment channel selection operation area, and an identity verification information input area are displayed in the cashier interface. For example, refer to a cashier interface shown in FIG. 2.

[0070] Step S122: The user performs operations such as payment channel selection and payment identity verification based on the cashier interface on the wallet side.

[0071] Step S123: The wallet app sends payment identity verification information and the like that are entered by the user to the wallet serving end.

[0072] Step S124: After verifying the payment identity verification information, the wallet serving end returns a payment result notification to the wallet app.

[0073] For example, if the verification on the payment identity verification information succeeds and a payment channel selected by the user is available, the returned payment result notification indicates payment success. Otherwise, payment failure is indicated. For example, refer to a payment result interface shown in FIG. 2.

[0074] The conventional payment link in the cross-border merchant-presented mode scenario is described above with reference to FIG. 1 and FIG. 2.

[0075] The applicant finds that in a process of providing a cross-border payment service based on the conventional payment link, in a phase from step S103 in which the user enters the amount and then taps to confirm to step S121 in which the cashier is opened (the user perceives a process from tapping to confirm based on the store page in FIG. 2 to seeing the cashier interface), average time consumption is relatively high, resulting in a large quantity of user losses. In addition, a

success rate is several points lower than that in a non-cross-border merchant-presented mode scenario, and even complaints from some users are received, resulting in poor user experience.

[0076] Further, the applicant finds a key problem by performing breakdown analysis on the conventional payment link: All steps in the conventional payment link are sequentially performed, that is, serially performed, and in particular, in steps S108 to S110, an interaction step of placing an order with the wallet serving end in the second region from the third-party platform in the first region is severely time-consuming.

[0077] Based on the above, the applicant proposes an improved solution to perform asynchronous processing on an order placement step involved in steps S108 to S110. Briefly, different from the conventional payment service link in which after placing an order with the wallet serving end in step S108, the third-party platform needs to wait for the wallet serving end to complete step S109 and then return the order placement response in step S110, and then performs step S111, in this improved solution, after performing step S108 to place an order with the wallet serving end, the third-party platform does not wait for completion of step S109 and step S110, but directly performs step S111 to deliver an order placement success notification to the regulatory platform. Therefore, by performing asynchronous order placement processing, a waiting time in an order placement phase can be omitted, to effectively reduce a payment time in the cross-border merchant-presented mode scenario.

[0078] Specific implementation steps of the above improved solution are further described below with reference to more accompanying drawings and embodiments. FIG. 3 is a schematic diagram of procedure steps of a processing method for a payment service according to an embodiment of this specification. Implementation of the method involves at least the following parts described above: a first acquiring institution and a third-party platform in a first region and a wallet serving end in a second region. The processing method for a payment service is applied to the third-party platform, or is performed by the third-party platform, and specifically includes the following steps.

[0079] First, in step S310, a third-party order number is generated based on a first order placement request received from an upstream platform, where the first order placement request includes information about a target payment service, and the target payment service is initiated by a user by scanning a collection two-dimensional code by using a wallet app corresponding to the wallet serving end.

[0080] It should be noted that the upstream platform is a platform located upstream of the third-party platform in a service link in which the user places an order with the wallet serving end by using the wallet app. In an embodiment, the upstream platform can be the regulatory platform shown in FIG. 1. In another embodiment, if there is no regulatory platform, the upstream platform can be an acquiring platform.

[0081] For example, in this step, content of the first order placement request and a timestamp of receiving the first order placement request can be concatenated, and then hash processing can be performed on content obtained through concatenation to serve as the third-party order number.

[0082] For descriptions of the first order placement request, the target payment service, and the like, refer to related content of the above descriptions of the conventional payment link, for example, related descriptions of step S101 to step S107. Details are not described.

[0083] Then, in step S320, a second order placement request is sent to the wallet serving end, and without waiting for return of an order placement success result corresponding to the second order placement request, a target notification indicating the order placement success result is directly sent to the upstream platform.

[0084] It should be understood that an asynchronous technology is used to implement this step. Asynchronization is a common term for asynchronous method calls, and refers to a program working mechanism in which a caller makes a return without waiting for method processing to be completed, and after the method processing is completed, a callback function registered in advance by the caller is called for subsequent processing.

[0085] Specifically, the generation of the third-party order number is performed through a main thread, and the method further includes: creating a first asynchronous task through the main thread, and submitting the first asynchronous task to a thread pool. Correspondingly, execution of this step can include: sending the second order placement request through a first thread scheduled by the thread pool for the first asynchronous task; and sending the target notification through the main thread.

[0086] Further, in an embodiment, the creating a first asynchronous task through the main thread can include: parsing the first order placement request to obtain product information corresponding to the target payment service, a to-be-paid first amount in a first currency, and an identifier of a login user in the wallet app; then converting the first amount into a second amount in a second currency based on a real-time exchange rate; and then determining context information

of the first asynchronous task based on the product information, the second amount, and the identifier of the login user. It should be understood that the context information is used when the thread pool calls the first thread to process the first asynchronous task.

[0087] In addition, in an embodiment, considering that a service involved by the third-party platform is usually not limited to a cross-regional merchant-presented mode scenario, for example, can be another service scenario such as a cross-border consumer-presented mode scenario, and in the another service scenario, there may be a unique service link that is not applicable to this improved solution, in this step, it can be first identified, based on the first order placement request, whether the target payment service is a cross-regional merchant-presented mode service. If it is identified that the target payment service is a cross-regional merchant-presented mode service, the first asynchronous task is created through the main thread. Otherwise, an original service procedure is maintained, for example, the main thread continues synchronous serial processing. It should be understood that if it is identified that the target payment service is a cross-regional merchant-presented mode service, to facilitate other subsequent processing, the third-party order number can be further marked as belonging to the cross-regional merchant-presented mode service.

[0088] For the sending of the target notification through the main thread, in an embodiment, the main thread can send the target notification in response to submitting the first asynchronous task to the thread pool. In addition, in an embodiment, for implementation of sending the target notification, for a to-be-filled target result record corresponding to the second order placement request in a database, a field indicating whether order placement at the wallet serving end succeeds can be filled with success, to automatically trigger sending of the target notification to the upstream platform. It should be understood that actually, the first asynchronous task is just submitted to the thread pool, and response data returned by the wallet serving end for the second order placement request is not received temporarily. In addition, another to-be-filled field (for example, a wallet-side order number and a third amount that actually needs to be paid) in the target result record is scheduled and filled by the thread pool instead of the main thread after it is subsequently detected that the response data is returned, so that the main thread is not interrupted, to implement true payment time omission instead of a time delay.

[0089] It should be noted that directly filling the order placement result field with success before the order placement response is returned is implemented based on the fact that a success rate of order placement at the wallet serving end in actual service statistics is 100% and a bottom

line strategy is preset on the third-party platform.

[0090] For step S310 and step S320, to further facilitate understanding, still from a perspective of execution by the third-party platform, a specific example is described below with reference to FIG. 4. Actually, in this improved solution, only service logic of the third-party platform needs to be improved.

[0091] As shown in FIG. 4, after processing order placement, the third-party platform generates the third-party order number for the target payment service in step S31, makes a scenario decision, identifies a cross-border merchant-presented mode service and a non-cross-border merchant-presented mode service, and marks only the identified cross-border merchant-presented mode service. Further, for an unmarked scenario, a wallet is directly and synchronously called based on an original procedure to place an order (for this, refer to steps S32a, S33a, and S34a).

[0092] For a marked scenario, step S32b is performed to perform asynchronous pre-processing, which is mainly to create the first asynchronous task, including preparing some key context information, for example, a wallet that is milk powder, a transaction amount, a transaction currency, and product information. After the task is constructed, step S33b is performed to submit the task to the asynchronous thread pool, and an order placement success notification is synchronously sent to the regulatory platform. Further, the thread pool asynchronously performs task scheduling. Order placement with the wallet is initiated in step S34b. In the first asynchronous task, after a result of the order placement with the wallet is obtained in step S36b, a callback is initiated. The order placement result is filled in step S37b, to store order placement response data in the database.

[0093] Finally, regardless of whether it is a synchronous or asynchronous procedure, in step S38, order placement post-processing is performed to uniformly update transaction information, a transaction status, and wallet information, and the overall procedure related to the order placement with the wallet is ended.

[0094] Asynchronous order placement can be implemented based on the above, and the target notification indicating the order placement success result is directly sent to the upstream platform before the order placement response is received.

[0095] Then, in step S330, an order number replacement request sent by the wallet serving end is received, where the order number replacement request carries an acquiring order number generated by the acquiring institution for the target payment service, and the acquiring order number is provided by the acquiring institution to the wallet app in response to the target

notification, and is sent by a third-party SDK embedded in the wallet app to the wallet serving end.

[0096] It should be noted that for descriptions of this step, refer to related descriptions of step S116 and steps S112 to S115 in FIG. 1. Details are not described herein.

[0097] Then, in step S340, target processing is performed on the order number replacement request, where the target processing includes: providing the third-party order number corresponding to the acquiring order number to the wallet serving end if the order placement success result is returned, so that the wallet serving end provides cashier interface information to the wallet app based on the third-party order number, to complete a target payment.

[0098] In an implementation A, the target processing includes: locating the corresponding target result record in the database based on the acquiring order number carried in the order number replacement request. As described above, the target result record is used to fill the order placement response data returned by the wallet serving end for the second order placement request (refer to step S320). It should be understood that there are a plurality of manners of locating the target result record. For example, the target result record includes the acquiring order number pre-filled by the third-party platform. In this case, the target result record can be directly located based on the acquiring order number.

[0099] Further, it can be queried, based on the located target result record, whether the order placement success result is returned or whether the order placement with the wallet is completed. It should be noted that the query is not implemented by querying a field value that is in the target result record and that indicates whether the order placement result is "success", because the field value is independently filled by the third-party platform before the order placement response is received, but is implemented by querying whether the target result record is fully filled, for example, whether a field corresponding to the wallet-side order number is not empty.

[0100] Then, the third-party order number corresponding to the acquiring order number is provided to the wallet serving end if it is found that the order placement with the wallet is completed. It should be understood that the acquiring order number and the third-party order number corresponding to the target payment service are associatively stored on the third-party platform in advance (refer to Table 1). Therefore, the corresponding third-party order number can be found based on the acquiring order number carried in the order number replacement request.

[0101] The implementation A is described above. In addition, the applicant considers that order number replacement depends on completion of the order placement with the wallet, and a service

interruption may occur in the implementation A, that is, the order number replacement arrives, but the order placement with the wallet is not completed.

[0102] To help understand the service interruption, the applicant draws FIG. 5. As shown in FIG. 5, in the above conventional payment link (marked as a current situation in FIG. 5), all phases of external order placement, order placement with the third-party platform, order placement with the wallet, SDK interception, order number replacement, and cashier rendering by the wallet are sequentially performed.

[0103] For an improved payment link obtained by introducing asynchronous order placement, in the implementation A, there are two possible optimized scenarios. In an optimized scenario 1, the order placement with the wallet is completed before order placement for replacement arrives, and there is no service loss in the scenario. In an optimized scenario 2, the order placement with the wallet is not completed before order placement for replacement arrives, that is, data of the order placement with the wallet is not stored in the database. In this case, search or return of the third-party order number is not triggered based on the service logic of the third-party platform, resulting in a service interruption and a service loss.

[0104] For the possible scenario 2, to ensure smooth completion of a payment procedure of the user, the applicant proposes another implementation B to introduce a retry mechanism into the target processing, so as to ensure smooth execution of the payment procedure of the user.

[0105] A difference between the implementation B and the implementation A is mainly as follows: In the implementation A, only one query is made and there is a service interruption if it is found that the order placement with the wallet is not completed. However, in the implementation B, if it is initially found that the order placement with the wallet is not completed, a retry task is created to make a retry query on whether the order placement with the wallet is completed, until it is found that the order placement success result is returned or the retry task expires.

[0106] For creation of the retry task, in an embodiment, if a result of the initial query is that the order placement with the wallet is not completed, it is determined whether the target payment service corresponding to the acquiring order number is a cross-regional merchant-presented mode service. If it is determined that the target payment service is a cross-regional merchant-presented mode service, the retry service is created. Otherwise, an original link procedure in a case of a corresponding service type continues to be performed. For example, it can be determined whether the third-party order number corresponding to the acquiring order number is marked as a cross-

regional merchant-presented mode service, to learn whether the target payment service is a cross-regional merchant-presented mode service. In another embodiment, it is assumed that the order number replacement request accepted on the third-party platform does not involve another service type. In this case, if it is initially found that the order placement with the wallet is not completed, the retry task can be directly created.

[0107] In addition, parameters that need to be configured when the retry task is created include an upper limit of a quantity of retries and waiting duration corresponding to respective retries. A timing start point corresponding to waiting duration for a first retry query is a point at which it is initially found that the order placement with the wallet is not completed.

[0108] In an embodiment, the applicant proposes to configure the waiting duration for the first retry query with reference to observed data of a service time in the conventional payment link, to ensure that a success rate of the first retry is 100% in a non-system exception case, so as to ensure system stability. Specifically, the configuring the waiting duration for the first retry can include:

[0109] (1) Time analysis of the order placement with the wallet (corresponding to steps S108 to S110)

[0110] An average time consumed to complete the order placement with the wallet in the conventional payment link can be counted, and then a time that can ensure experience of most users can be determined based on the average time, and is referred to as a first statistical time in this specification. For example, a curve is drawn for the average time for the order placement with the wallet by using a weekly statistical granularity. It can be seen that the average time is 280 ms, and there are some spikes, but the percentage of spikes is very low. To ensure an acceleration effect of 80% of transactions, a P80 time, that is, 330 ms, is used as a guarantee baseline (or referred to as the first statistical time). This represents the experience of most users.

[0111] (2) Time analysis of SDK interception (corresponding to steps S111 to S116)

[0112] A shortest time for SDK interception can be determined as a second statistical time. For example, it can be seen by drawing a histogram that transactions with a time falling within an interval [0 ms, 300 ms] account for 1%, and transactions with a time greater than 300 ms account for 99%. Considering that 300 ms exceeds the average time of 280 ms, this means that essentially only 1% of transactions with a time less than 300 ms need to be protected by the retry mechanism.

[0113] (3) Service objective and strategy

[0114] The service objective can be as follows: Only one retry is made for a transaction in the

non-system exception case. For example, in the above example, only one retry needs to be made for a transaction whose SDK interception time is 1%. Therefore, the waiting duration for the first retry can be a calculation result obtained by subtracting the second statistical time from the first statistical time, for example, 330 ms–300 ms=30 ms.

[0115] In addition, considering that compared with that in the conventional payment link, a scheduling time needs to be additionally consumed from submission of the first asynchronous task involved in the order placement with the wallet to the thread pool to calling of the first thread by the thread pool to execute the first asynchronous task, it is proposed to consider an average time, which is referred to as a third statistical time in this specification, consumed for scheduling an asynchronous task. In this case, the waiting duration for the first retry can be a calculation result obtained by subtracting the second statistical time from a sum of the first statistical time and the third statistical time, for example, (330 ms+10 ms)–300 ms=40 ms.

[0116] The logic and strategy for configuring the waiting duration for the first retry to ensure that the success rate of the first retry is 100% in the non-system exception case and to minimize the waiting duration.

[0117] In another embodiment, to ensure retry success in a system exception case, a plurality of retries can be designed, and related parameters can be configured by using an exponential backoff algorithm. It should be understood that the exponential backoff algorithm is an error recovery mechanism, and is usually used to control retries in network communication congestion. This algorithm reduces a retry frequency by gradually increasing a waiting time, to alleviate a load on a server or network, avoid congestion caused by excessive concurrent attempts, and give a system enough time to restore to a normal state. Specifically, for technical parameters of the plurality of retries, there are the following analysis strategies:

[0118] ① Retry mechanism: The exponential backoff algorithm Math.pow(2, quantity of retries)*retry frequency is used, that is, each retry interval is twice a previous retry interval, and a retry period is gradually extended to avoid pressure on the system caused by frequent retries.

[0119] ② Retry frequency: To ensure a first service retry objective, the retry frequency can be set to the waiting duration for the first retry, for example, 30 ms or 40 ms.

[0120] ③ Quantity of retries: The quantity of retries can be set with reference to a threshold for application service expiration. For example, if the threshold for application service expiration is 3s, and 40+80+160+320+640+1280=2520 ms. Therefore, a maximum of six retries can be made.

[0121] The parameters that need to be configured when the retry task is created are described above.

[0122] In addition, the created retry task can be an asynchronous retry task, so that during a retry, the main thread is not blocked and can continue to execute other tasks or process other events.

[0123] The target processing can be implemented on the order number replacement request based on the above, so that the found third-party order number is returned to the wallet serving end if the order placement with the wallet is completed, to facilitate completion of the target payment. It should be understood that for a procedure after the wallet serving end obtains the third-party order number, refer to the above descriptions of steps S119 to S124 in FIG. 1. Details are not described herein.

[0124] In conclusion, according to the processing method for a payment service disclosed in this embodiment of this specification, asynchronous processing is performed on order placement with a wallet, to effectively reduce a payment time in a cross-regional merchant-presented mode service scenario. Further, an asynchronous retry for order number replacement is introduced, to resolve a possible problem that procedure asynchronization affects a payment success rate, and achieve a win-win situation for a payment time and a service success rate. Further, an automated retry parameter configuration manner is formed in the asynchronous retry, to ensure 100% success of a first retry in a non-system exception case. In addition, system stability is ensured by using a plurality of retry parameters designed for a system exception case.

[0125] FIG. 6 is a schematic structural diagram of a processing apparatus for a payment service integrated into a third-party platform according to an embodiment of this specification. The payment service involves at least an acquiring institution and a third-party platform in a first region and a wallet serving end in a second region. The apparatus is integrated into the third-party platform. As shown in FIG. 6, the apparatus 600 includes:

a third-party order number generation module 610, configured to generate a third-party order number based on a first order placement request received from an upstream platform, where the first order placement request includes information about a target payment service, and the target payment service is initiated by a user by scanning a collection two-dimensional code by using a wallet app corresponding to the wallet serving end; an asynchronous processing module 620, configured to send a second order placement request to the wallet serving end, and without waiting for return of an order placement success result corresponding to the second order placement request, directly send a target notification indicating the order placement success result

to the upstream platform; a replacement request receiving module 630, configured to receive an order number replacement request sent by the wallet serving end, where the order number replacement request carries an acquiring order number generated by the acquiring institution for the target payment service, and the acquiring order number is provided by the acquiring institution to the wallet app in response to the target notification, and is sent by a third-party SDK embedded in the wallet app to the wallet serving end; and a target processing module 640, configured to perform target processing on the order number replacement request, where the target processing includes: providing the third-party order number corresponding to the acquiring order number to the wallet serving end if the order placement success result is returned, so that the wallet serving end provides cashier interface information to the wallet app based on the third-party order number, to complete a target payment.

[0126] In an embodiment, the generating a third-party order number is performed through a main thread, and the asynchronous processing module 620 is further configured to create a first asynchronous task through the main thread, and submit the first asynchronous task to a thread pool; and that an asynchronous processing module 620 is configured to send a second order placement request to the wallet serving end, and without waiting for return of an order placement success result corresponding to the second order placement request, directly send a target notification indicating the order placement success result to the upstream platform specifically includes: sending the second order placement request through a first thread scheduled by the thread pool for the first asynchronous task; and sending the target notification through the main thread.

[0127] In a specific embodiment, that the asynchronous processing module 620 is configured to create a first asynchronous task through the main thread, and submit the first asynchronous task to a thread pool specifically includes: if it is identified that the target payment service is a cross-regional merchant-presented mode service, creating the first asynchronous task through the main thread, and submitting the first asynchronous task to the thread pool.

[0128] In a specific embodiment, official currencies in the first region and the second region are respectively a first currency and a second currency, and that the asynchronous processing module 620 is configured to create a first asynchronous task through the main thread specifically includes: parsing the first order placement request to obtain product information corresponding to the target payment service, a to-be-paid first amount in the first currency, and an identifier of a login user in the wallet app; converting the first amount into a second amount in the second currency based on a real-time exchange rate; and determining context information of the first

asynchronous task based on the product information, the second amount, and the identifier of the login user.

[0129] In an embodiment, the collection two-dimensional code is an aggregation code.

[0130] In an embodiment, the upstream platform is the acquiring institution or a regulatory platform.

[0131] In an embodiment, that an asynchronous processing module 620 is configured to directly send a target notification indicating the order placement success result to the upstream platform specifically includes: for a to-be-filled target result record corresponding to the second order placement request in a database, filling a field indicating whether order placement succeeds with success, to automatically trigger sending of the target notification to the upstream platform.

[0132] In a specific embodiment, the asynchronous processing module 620 is further configured to fully fill the target result record by using a parsing result of order placement response data in response to detecting an event that the wallet serving end returns an order placement response, where the parsing result includes the order placement success result.

[0133] In a specific embodiment, the target processing module 640 is specifically configured to provide the third-party order number to the wallet serving end if it is found that the target result record is fully filled.

[0134] In an embodiment, the target processing module 640 is configured to make, based on the acquiring order number carried in the order number replacement request, an initial query on whether the order placement success result is returned; create a retry task if a result of the initial query is "not returned"; and in a process of executing the retry task, make, based on a predetermined time interval, a retry query on whether the order placement success result is returned until it is found that the order placement success result is returned or the task expires.

[0135] In a specific embodiment, the target processing module 640 is specifically configured to determine whether the target payment service is a cross-regional merchant-presented mode service if the result of the initial query is "not returned"; and create the retry task if it is determined that the target payment service is a cross-regional merchant-presented mode service.

[0136] Further, in an example, a setting basis for waiting duration before a first retry in the retry task includes a first statistical time and a second statistical time corresponding to a first link and a second link in parallel, the first link corresponds to a link from sending of the second order placement request to receiving of a returned order placement response by the third-party platform,

and the second link corresponds to a link from sending of the target notification to receiving of the order number replacement request by the third-party platform.

[0137] In a specific example, the sending of the second order placement request is performed by scheduling a first asynchronous task created for the target payment service, and the setting basis further includes a third statistical time consumed for scheduling and executing the first asynchronous task; and the waiting duration is a value obtained by adding the first statistical time and the third statistical time and then subtracting the second statistical time.

[0138] In a specific example, the predetermined time interval is configured by using an exponential backoff algorithm, and a retry frequency in the algorithm is set to the waiting duration.

[0139] For ease of description, the above apparatus is described by dividing the apparatus into various modules based on functions. Certainly, during implementation of this specification, functions of all modules can be implemented in one or more pieces of software and/or hardware.

[0140] The specification also provides a computer-readable non-transitory storage medium, wherein the storage medium stores a computer program, which, when executed by a processor, can be used to perform one or more steps of one or more methods described or illustrated herein or provides functionality described or illustrated herein. Herein, the computer-readable non-transitory storage medium or media may include one or more semiconductor-based or other integrated circuits (ICs) (such, as for example, field-programmable gate arrays (FPGAs) or application-specific ICs (ASICs)), hard disk drives (HDDs), hybrid hard drives (HHDs), optical discs, optical disc drives (ODDs), magneto-optical discs, magneto-optical drives, floppy diskettes, floppy disk drives (FDDs), magnetic tapes, solid-state drives (SSDs), RAM-drives, SECURE DIGITAL cards or drives, any other suitable computer-readable non-transitory storage media, or any suitable combination of two or more of these, where appropriate. A computer-readable non-transitory storage medium may be volatile, non-volatile, or a combination of volatile and non-volatile, where appropriate.

[0141] The specification also provides a computer system. FIG. 7 illustrates an example computer system 700. In particular embodiments, one or more computer systems 700 perform one or more steps of one or more methods described or illustrated herein. In particular embodiments, one or more computer systems 700 provide functionality described or illustrated herein. In particular embodiments, software running on one or more computer systems 700 performs one or more steps of one or more methods described or illustrated herein or provides functionality

described or illustrated herein. Particular embodiments include one or more portions of one or more computer systems 700. Herein, reference to a computer system may encompass a computing device, and vice versa, where appropriate. Moreover, reference to a computer system may encompass one or more computer systems, where appropriate.

[0142] This disclosure contemplates any suitable number of computer systems 700. This disclosure contemplates computer system 700 taking any suitable physical form. As example and not by way of limitation, computer system 700 may be an embedded computer system, a system-on-chip (SOC), a single-board computer system (SBC) (such as, for example, a computer-on-module (COM) or system-on-module (SOM)), a desktop computer system, a laptop or notebook computer system, an interactive kiosk, a mainframe, a mesh of computer systems, a mobile telephone, a personal digital assistant (PDA), a server, a tablet computer system, or a combination of two or more of these. Where appropriate, computer system 700 may include one or more computer systems 700; be unitary or distributed; span multiple locations; span multiple machines; span multiple data centers; or reside in a cloud, which may include one or more cloud components in one or more networks. Where appropriate, one or more computer systems 700 may perform without substantial spatial or temporal limitation one or more steps of one or more methods described or illustrated herein. As an example and not by way of limitation, one or more computer systems 700 may perform in real time or in batch mode one or more steps of one or more methods described or illustrated herein. One or more computer systems 700 may perform at different times or at different locations one or more steps of one or more methods described or illustrated herein, where appropriate.

[0143] In particular embodiments, computer system 700 includes a processor 702, memory 704, storage 706, an input/output (I/O) interface 708, a communication interface 710, and a bus 712. Although this disclosure describes and illustrates a particular computer system having a particular number of particular components in a particular arrangement, this disclosure contemplates any suitable computer system having any suitable number of any suitable components in any suitable arrangement.

[0144] In particular embodiments, processor 702 includes hardware for executing instructions, such as those making up a computer program. As an example and not by way of limitation, to execute instructions, processor 702 may retrieve (or fetch) the instructions from an internal register, an internal cache, memory 704, or storage 706; decode and execute them; and then write one or

more results to an internal register, an internal cache, memory 704, or storage 706. In particular embodiments, processor 702 may include one or more internal caches for data, instructions, or addresses. This disclosure contemplates processor 702 including any suitable number of any suitable internal caches, where appropriate. As an example and not by way of limitation, processor 702 may include one or more instruction caches, one or more data caches, and one or more translation lookaside buffers (TLB s). Instructions in the instruction caches may be copies of instructions in memory 704 or storage 706, and the instruction caches may speed up retrieval of those instructions by processor 702. Data in the data caches may be copies of data of 80 in memory 704 or storage 706 for instructions executing at processor 702 to operate on; the results of previous instructions executed at processor 702 for access by subsequent instructions executing at processor 702 or for writing to memory 704 or storage 706; or other suitable data. The data caches may speed up read or write operations by processor 702. The TLBs may speed up virtual-address translation for processor 702. In particular embodiments, processor 702 may include one or more internal registers for data, instructions, or addresses. This disclosure contemplates processor 702 including any suitable number of any suitable internal registers, where appropriate. Where appropriate, processor 702 may include one or more arithmetic logic units (ALUs); be a multi-core processor; or include one or more processors 702. Although this disclosure describes and illustrates a particular processor, this disclosure contemplates any suitable processor.

[0145] In particular embodiments, memory 704 includes main memory for storing instructions for processor 702 to execute or data for processor 702 to operate on. As an example and not by way of limitation, computer system 700 may load instructions from storage 706 or another source (such as, for example, another computer system 700) to memory 704. Processor 702 may then load the instructions from memory 704 to an internal register or internal cache. To execute the instructions, processor 702 may retrieve the instructions from the internal register or internal cache and decode them. During or after execution of the instructions, processor 702 may write one or more results (which may be intermediate or final results) to the internal register or internal cache. Processor 702 may then write one or more of those results to memory 704. In particular embodiments, processor 702 executes only instructions in one or more internal registers or internal caches or in memory 704 (as opposed to storage 706 or elsewhere) and operates only on data in one or more internal registers or internal caches or in memory 704 (as opposed to storage 706 or elsewhere). One or more memory buses (which may each include an address bus and a data bus)

may couple processor 702 to memory 704. Bus 712 may include one or more memory buses, as described below. In particular embodiments, one or more memory management units (MMUs) reside between processor 702 and memory 704 and facilitate accesses to memory 704 requested by processor 702. In particular embodiments, memory 704 includes random access memory (RAM). This RAM may be volatile memory, where appropriate. Where appropriate, this RAM may be dynamic RAM (DRAM) or static RAM (SRAM). Moreover, where appropriate, this RAM may be single-ported or multi-ported RAM. This disclosure contemplates any suitable RAM. Memory 704 may include one or more memories 704, where appropriate. Although this disclosure describes and illustrates particular memory, this disclosure contemplates any suitable memory.

[0146] In particular embodiments, storage 706 includes mass storage for data or instructions. As an example and not by way of limitation, storage 706 may include a hard disk drive (HDD), a floppy disk drive, flash memory, an optical disc, a magneto-optical disc, magnetic tape, or a Universal Serial Bus (USB) drive or a combination of two or more of these. Storage 706 may include removable or non-removable (or fixed) media, where appropriate. Storage 706 may be internal or external to computer system 700, where appropriate. In particular embodiments, storage 706 is non-volatile, solid-state memory. In particular embodiments, storage 706 includes read-only memory (ROM). Where appropriate, this ROM may be mask-programmed ROM, programmable ROM (PROM), erasable PROM (EPROM), electrically erasable PROM (EEPROM), electrically alterable ROM (EAROM), or flash memory or a combination of two or more of these. This disclosure contemplates mass storage 706 taking any suitable physical form. Storage 706 may include one or more storage control units facilitating communication between processor 702 and storage 706, where appropriate. Where appropriate, storage 706 may include one or more storages 706. Although this disclosure describes and illustrates particular storage, this disclosure contemplates any suitable storage.

[0147] In particular embodiments, I/O interface 708 includes hardware, software, or both, providing one or more interfaces for communication between computer system 700 and one or more I/O devices. Computer system 700 may include one or more of these I/O devices, where appropriate. One or more of these I/O devices may enable communication between a person and computer system 700. As an example and not by way of limitation, an I/O device may include a keyboard, keypad, microphone, monitor, mouse, printer, scanner, speaker, still camera, stylus, tablet, touch screen, trackball, video camera, another suitable I/O device or a combination of two

or more of these. An I/O device may include one or more sensors. This disclosure contemplates any suitable I/O devices and any suitable I/O interfaces 708 for them. Where appropriate, I/O interface 708 may include one or more device or software drivers enabling processor 702 to drive one or more of these I/O devices. I/O interface 708 may include one or more I/O interfaces 708, where appropriate. Although this disclosure describes and illustrates a particular I/O interface, this disclosure contemplates any suitable I/O interface.

[0148] In particular embodiments, communication interface 710 includes hardware, software, or both providing one or more interfaces for communication (such as, for example, packet-based communication) between computer system 700 and one or more other computer systems 700 or one or more networks. As an example and not by way of limitation, communication interface 710 may include a network interface controller (NIC) or network adapter for communicating with an Ethernet or other wire-based network or a wireless NIC (WNIC) or wireless adapter for communicating with a wireless network, such as a WI-FI network. This disclosure contemplates any suitable network and any suitable communication interface 710 for it. As an example and not by way of limitation, computer system 700 may communicate with an ad hoc network, a personal area network (PAN), a local area network (LAN), a wide area network (WAN), a metropolitan area network (MAN), or one or more portions of the Internet or a combination of two or more of these. One or more portions of one or more of these networks may be wired or wireless. As an example, computer system 700 may communicate with a wireless PAN (WPAN) (such as, for example, a BLUETOOTH WPAN), a WI-FI network, a WI-MAX network, a cellular telephone network (such as, for example, a Global System for Mobile Communications (GSM) network), or other suitable wireless network or a combination of two or more of these. Computer system 700 may include any suitable communication interface 710 for any of these networks, where appropriate. Communication interface 710 may include one or more communication interfaces 710, where appropriate. Although this disclosure describes and illustrates a particular communication interface, this disclosure contemplates any suitable communication interface.

[0149] In particular embodiments, bus 712 includes hardware, software, or both coupling components of computer system 700 to each other. As an example and not by way of limitation, bus 712 may include an Accelerated Graphics Port (AGP) or other graphics bus, an Enhanced Industry Standard Architecture (EISA) bus, a front-side bus (FSB), a HYPERTRANSPORT (HT) interconnect, an Industry Standard Architecture (ISA) bus, an INFINIBAND interconnect, a low-

pin-count (LPC) bus, a memory bus, a Micro Channel Architecture (MCA) bus, a Peripheral Component Interconnect (PCI) bus, a PCI-Express (PCIe) bus, a serial advanced technology attachment (SATA) bus, a Video Electronics Standards Association local (VLB) bus, or another suitable bus or a combination of two or more of these. Bus 712 may include one or more buses 712, where appropriate. Although this disclosure describes and illustrates a particular bus, this disclosure contemplates any suitable bus or interconnect.

[0150] A person skilled in the art should be aware that in the above one or more examples, functions described in this application can be implemented by hardware, software, firmware, or any combination thereof. When being implemented by software, these functions can be stored in a computer-readable medium or transmitted as one or more instructions or code on a computer-readable medium.

[0151] In the above specific implementations, the objectives, technical solutions, and beneficial effects of the present invention are further described in detail. It should be understood that the above descriptions are merely specific implementations of the present invention, but are not intended to limit the protection scope of the present invention. Any modification, equivalent replacement, improvement, and the like made based on the technical solutions of the present invention shall fall within the protection scope of the present invention.

CLAIMS

What is claimed is:

1. A processing method for a payment service that involves at least an acquiring institution and a third-party platform in a first region and a wallet serving end in a second region, wherein the method is applied to the third-party platform and comprises:

generating a third-party order number based on a first order placement request received from an upstream platform, wherein the first order placement request comprises information about a target payment service, and the target payment service is initiated by a user by scanning a collection two-dimensional code by using a wallet app corresponding to the wallet serving end;

sending a second order placement request to the wallet serving end, and without waiting for return of an order placement success result corresponding to the second order placement request, directly sending a target notification indicating the order placement success result to the upstream platform;

receiving an order number replacement request sent by the wallet serving end, wherein the order number replacement request carries an acquiring order number generated by the acquiring institution for the target payment service, and the acquiring order number is provided by the acquiring institution to the wallet app in response to the target notification, and is sent by a third-party SDK embedded in the wallet app to the wallet serving end; and

performing target processing on the order number replacement request, wherein the target processing comprises: providing the third-party order number corresponding to the acquiring order number to the wallet serving end if the order placement success result is returned, so that the wallet serving end provides cashier interface information to the wallet app based on the third-party order number, to complete a target payment.

2. The method according to claim 1, wherein the generating a third-party order number is performed through a main thread, and the method further comprises:

creating a first asynchronous task through the main thread, and submitting the first asynchronous task to a thread pool; and

the sending a second order placement request to the wallet serving end, and without waiting for return of an order placement success result corresponding to the second order placement request, directly sending a target notification indicating the order placement success result to the

upstream platform comprises:

sending the second order placement request through a first thread scheduled by the thread pool for the first asynchronous task; and

sending the target notification through the main thread.

3. The method according to claim 2, wherein the creating a first asynchronous task through the main thread, and submitting the first asynchronous task to a thread pool comprises:

if it is identified that the target payment service is a cross-regional merchant-presented mode service, creating the first asynchronous task through the main thread, and submitting the first asynchronous task to the thread pool.

4. The method according to claim 2, wherein official currencies in the first region and the second region are respectively a first currency and a second currency, and the creating a first asynchronous task through the main thread comprises:

parsing the first order placement request to obtain product information corresponding to the target payment service, a to-be-paid first amount in the first currency, and an identifier of a login user in the wallet app;

converting the first amount into a second amount in the second currency based on a real-time exchange rate; and

determining context information of the first asynchronous task based on the product information, the second amount, and the identifier of the login user.

5. The method according to claim 1, wherein the collection two-dimensional code is an aggregation code.

6. The method according to claim 1, wherein the upstream platform is the acquiring institution or a regulatory platform.

7. The method according to claim 1, wherein the directly sending a target notification indicating the order placement success result to the upstream platform comprises:

for a to-be-filled target result record corresponding to the second order placement request in a database, filling a field indicating whether order placement succeeds with success, to automatically trigger sending of the target notification to the upstream platform.

8. The method according to claim 7, wherein before the performing target processing on the order number replacement request, the method further comprises:

fully filling the target result record by using a parsing result of order placement response data

in response to detecting an event that the wallet serving end returns an order placement response, wherein the parsing result comprises the order placement success result.

9. The method according to claim 7, wherein the providing the third-party order number corresponding to the acquiring order number to the wallet serving end if the order placement success result is returned comprises:

providing the third-party order number to the wallet serving end if it is found that the target result record is fully filled.

10. The method according to claim 1, wherein the target processing comprises:

making, based on the acquiring order number carried in the order number replacement request, an initial query on whether the order placement success result is returned;

creating a retry task if a result of the initial query is "not returned"; and

in a process of executing the retry task, making, based on a predetermined time interval, a retry query on whether the order placement success result is returned until it is found that the order placement success result is returned or the task expires.

11. The method according to claim 10, wherein the creating a retry task if a result of the initial query is "not returned" comprises:

determining whether the target payment service is a cross-regional merchant-presented mode service if the result of the initial query is "not returned"; and

creating the retry task if it is determined that the target payment service is a cross-regional merchant-presented mode service.

12. The method according to claim 11, wherein a setting basis for waiting duration before a first retry in the retry task comprises a first statistical time and a second statistical time corresponding to a first link and a second link in parallel, the first link corresponds to a link from sending of the second order placement request to receiving of a returned order placement response by the third-party platform, and the second link corresponds to a link from sending of the target notification to receiving of the order number replacement request by the third-party platform.

13. The method according to claim 12, wherein the sending of the second order placement request is performed by scheduling a first asynchronous task created for the target payment service, and the setting basis further comprises a third statistical time consumed for scheduling and executing the first asynchronous task.

14. The method according to claim 13, wherein the waiting duration is a value obtained by

adding the first statistical time and the third statistical time and then subtracting the second statistical time.

15. The method according to claim 14, wherein the predetermined time interval is configured by using an exponential backoff algorithm, and a retry frequency in the algorithm is set to the waiting duration.

16. A computer system, comprising a memory, a processor, and a computer program that is stored in the memory and that is capable of running on the processor, wherein the processor is integrated into a third-party platform, the third-party platform and an acquiring institution are located in a first region, and when the processor executes the program, the following steps are implemented:

generating a third-party order number based on a first order placement request received from an upstream platform, wherein the first order placement request comprises information about a target payment service, the target payment service is initiated by a user by scanning a collection two-dimensional code by using a wallet app corresponding to a wallet serving end, and the wallet serving end is located in a second region;

sending a second order placement request to the wallet serving end, and without waiting for return of an order placement success result corresponding to the second order placement request, directly sending a target notification indicating the order placement success result to the upstream platform;

receiving an order number replacement request sent by the wallet serving end, wherein the order number replacement request carries an acquiring order number generated by the acquiring institution for the target payment service, and the acquiring order number is provided by the acquiring institution to the wallet app in response to the target notification, and is sent by a third-party SDK embedded in the wallet app to the wallet serving end; and

performing target processing on the order number replacement request, wherein the target processing comprises: providing the third-party order number corresponding to the acquiring order number to the wallet serving end if the order placement success result is returned, so that the wallet serving end provides cashier interface information to the wallet app based on the third-party order number, to complete a target payment.

17. A processing apparatus for a payment service, wherein the payment service involves at least an acquiring institution and a third-party platform in a first region and a wallet serving end

in a second region, and the apparatus is integrated into the third-party platform and comprises:

a third-party order number generation module, configured to generate a third-party order number based on a first order placement request received from an upstream platform, wherein the first order placement request comprises information about a target payment service, and the target payment service is initiated by a user by scanning a collection two-dimensional code by using a wallet app corresponding to the wallet serving end;

an asynchronous processing module, configured to send a second order placement request to the wallet serving end, and without waiting for return of an order placement success result corresponding to the second order placement request, directly send a target notification indicating the order placement success result to the upstream platform;

a replacement request receiving module, configured to receive an order number replacement request sent by the wallet serving end, wherein the order number replacement request carries an acquiring order number generated by the acquiring institution for the target payment service, and the acquiring order number is provided by the acquiring institution to the wallet app in response to the target notification, and is sent by a third-party SDK embedded in the wallet app to the wallet serving end; and

a target processing module, configured to perform target processing on the order number replacement request, wherein the target processing comprises: providing the third-party order number corresponding to the acquiring order number to the wallet serving end if the order placement success result is returned, so that the wallet serving end provides cashier interface information to the wallet app based on the third-party order number, to complete a target payment.

18. A computer-readable nonvolatile storage medium, wherein the storage medium stores a computer program, and when the computer program is executed by a processor, the method according to any one of claims 1 to 15 is implemented.