PAYMENT GATEWAY SPECIFICATIONS

Mohamed El-Zahaby
V 0.1
August 2021

This guide is primarily for technical teams engaged in developing a payment gateway, rather than for those using an existing one. It offers an extensive introduction to the business aspects of payment gateways. The document provides critical insights into both the business and technical facets necessary for constructing a payment gateway. It includes a detailed exploration of MPGS integration within the Connector Layer. This document marks the beginning of a series aimed at developing a comprehensive White Label Payment Gateway. It's important to note that much of the content is compiled from various online resources, reflecting the extensive effort in ..curating and organizing this information from numerous sources

Contents

INTRODUCTION

What is a Payment Gateway?

Payment Gateway is a software that works as the middleman (middleware layer) between customers (card-holders) (payers) and the merchants to process payments for online purchases.

How does Payment Gateway work?

- Allow merchants to add different payment methods.
- To ensure the security of the transactions.
- Transfer the card data to the bank.
- Send responses back to the Merchant and customer.
- Provide different dashboards and reports for the merchants.
- Allow the merchants to settle and conciliate their balances with the bank easily.

What are the key players in an online payment operation?

Once a customer clicks the **Purchase** button on a merchant's website, there are some key players involved in such process:

- Merchant: the online business owner operates in any vertical (retail, tickets, eCommerce, online gaming, etc.), offering products or services to customers.

- Card-Holder (Customer): the card-holder, also called a customer, who wants to buy (or pay for) the products or services that the Merchant is selling. The customer is the player who initiates the transaction cycle.

- Acquirer: The acquiring bank is the financial institution that maintains the Merchant's bank account (known as the Merchant's account). The acquiring bank passes the Merchant's transactions to the issuing bank to receive payment

- Issuer: also known as the Issuing bank, this issuer issues the cardholder's credit or debit card on behalf of the card schemes (Visa, Mastercard). These banks are responsible for generating a monthly statement, collecting the monthly bill, putting late payment fines and helping you with all credit card-related issues. It extends a line of credit to the consumer. Liability for non-payment is then shared by the issuing bank and the acquiring bank, according to rules established by the card

association brand.

- Payment Network: also known as card-network, Payment Networks are the companies printed on your credit card (typically branded with a MasterCard, VISA, or American Express, etc., logo). These companies have deals with different banks, this allows a card issued by Domestic Bank in USA to be used at a shop in Egypt, for instance, without requiring the banks to have a direct relationship with each other. And this network facilitates the payment transaction between the Merchant and issuer, i.e., the customer (card-holder) to the Merchant to acquiring-bank and source issuing bank of funds.

What is MID (bank merchant account)?

To build your payment gateway, you have to own a merchant account at a bank (acquirer bank).
You request a merchant account from the bank. The bank checks the legalities and validity; once your application for a merchant account is approved from CBE (Central Bank of Egypt), you will be assigned a merchant identification number (MID).
MID is an account identification for your merchant account. It is required to process card transactions and move funds from your customers' issuing-banks accounts to your acquiring-bank account once their payments are authorized and ready to be settled by your acquirer.

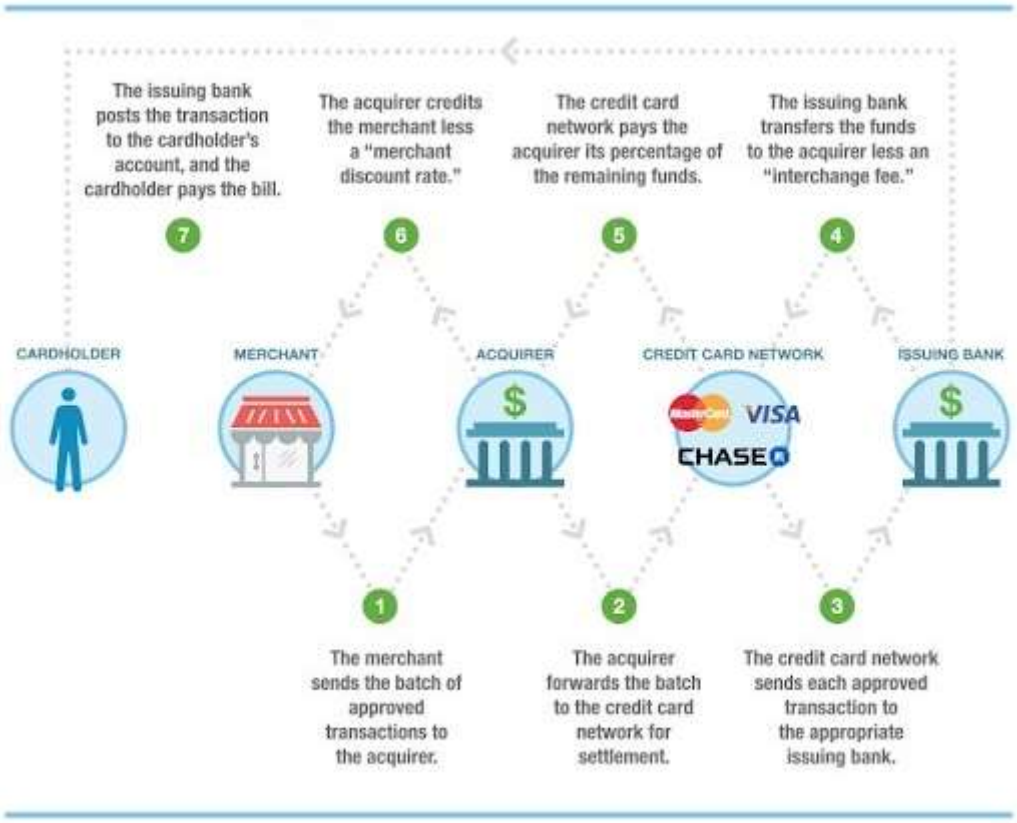
Why do we use merchant-account in the payment process? And why can't the money be transferred directly from the issuing bank into the seller's bank account?

Merchant accounts are necessary to maintain the chain of approval from when your customer submits their card details to when you receive the money.
Since merchandise can be returned, there's always the chance that some money you receive as an online seller will have to be paid back due to returns or chargebacks. Returns account for a certain level of risk in your transactions, depending on the vertical you're operating and the nature of your business. The acquirer subtracts returns from the funds sitting in the merchant account at that given time; then, they transfer the remaining funds to your business bank account.
On top of that, your payment gateway may be accumulating deposits from multiple sources. A merchant account simplifies how you're getting paid; your payment gateway collects them in your merchant account and combines them into one single deposit for your bank account, making reconciliation easier. Payment reconciliation is the process of

checking your bank statements against your accounting and your payment gateway, PSP, ISO or acquirer records to ensure the payment amounts match.

What is the cycle of the card payment process?

To build your payment gateway, you have to own a merchant account at a bank (acquirer bank).



There is a series of events behind any successful transaction:

- The **Card-Holder** purchases a Mobile phone of 100 EGP through the **Merchant (Seller)** online store using the **Payment-Gateway**.
- The **Payment-Gateway** calls the **Acquirer-Bank** to validate the MID.
- The **Acquirer-Bank** calls the **Card-Network** (card schemes) to validate the card's details.
- In case of using 3D-Secure, the **Card-Network** (card schemes) sends OTP to the **Card-Holder** to verify the card.
- The **Card-Network** (card schemes) calls the Issuer-Bank to validate the balance. And request 98.50 EGP (for example) from the **Issuer-Bank**. Here the **Issuer-Bank**

gains 1.50 EGP as **Interchange fees** (Interchange fees are determined by many complex variables. In simple terms, it is a flat rate plus a percentage of the total sales value).

- The **Issuer-Bank** approves and pays 98.50 EGP (for example) to the **Card-Network** (card schemes).

- The **Card-Network** (card schemes) keeps (gains) 0.50 EGP (for example) and sends 98 EGP (for example) to the **Acquirer-Bank**; this is the **Network Fee**

- The **Acquirer-Bank** keeps (gains) 1.00 EGP (for example) and sends 97 EGP (for example) to the Payment-Gateway account; this is the **Acquirer fees**.

- The **Payment-Gateway** keeps (gains) 0.50 EGP (for example) and sends 96.5 EGP (for example) to the **Merchant (Seller)** account.

In Short, Merchant bears the overall loss, and his discount is distributed between Acquiring Bank (Acquirer fees), Issuing Bank (Interchange Fees), and Payment Network (Network Fee). No doubt, these Card companies are investing too much in promoting their card and encouraging people for cashless transactions.

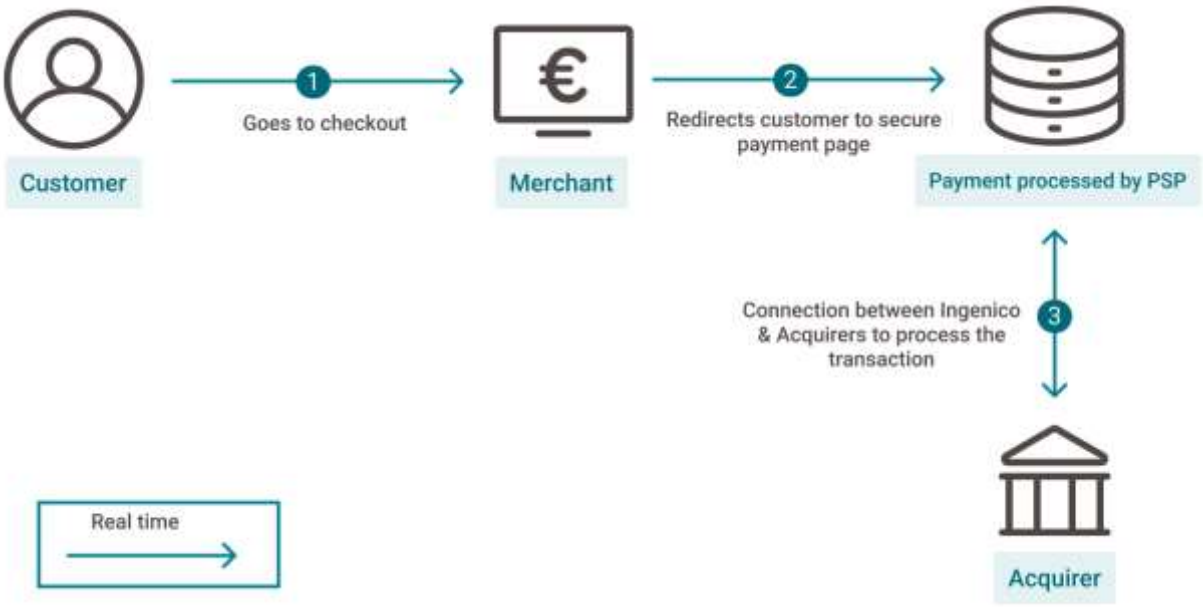
How does a payment gateway work throughout the payment journey?

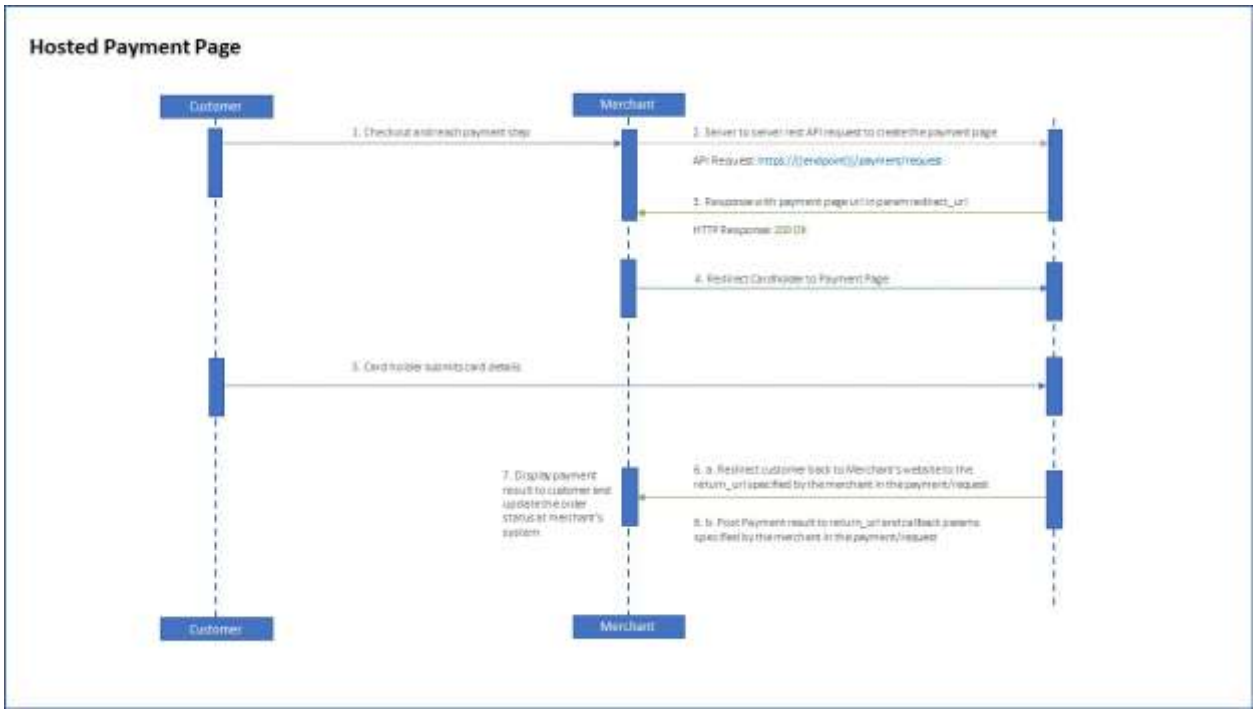
The steps below are the typical steps across the different exist different online payment gateways. It may differ or vary from one payment gateway to the other based on the business needs.

1- The customer chooses the product or service they want to purchase and proceeds to the payment page. Most payment gateways offer you different options for your payment page. The common payment methods are the following:

Hosted payment page (HPP) / Hosted Checkout

A hosted payment page is an out-of-the-box payment page where customers are redirected when they are ready to checkout. The payment gateway securely receives the transaction data before it passes it to the acquirer. A hosted payment page reduces the PCI burden for online merchants if you don’t collect and/ or store the card-holder data on your server.





The pros of a hosted payment gateway are that all payment processing is taken by the service provider. Client card data is also stored by the vendor. So using a hosted gateway requires no PCI compliance and offers pretty easy integration.

The cons are that there is a lack of control over a hosted gateway. Customers may not trust third-party payment systems. Besides that, redirecting them away from your website lowers conversion rate and doesn't help your branding either.

Server-to-server integration (API) / (Direct Integration)

A server to server integration is also known as a direct integration as it enables the communication between two servers; the Merchant's server with the payment gateway's server. By requesting the card details on the payment page, a direct transaction can be initiated. Customers can finalize a card payment without being redirected to the payment

page of the payment gateway, resulting in faster checkout, a more consistent user experience, and more control over the look and feel of the payment page from the Merchant's perspective. A server-to-server integration is suitable if you collect and/ or store the payment data before sending them to the payment gateway for processing.

The pros of this method are equal to an integrated payment gateway. You get the customization options and branding capabilities without PCI DSS compliance. The user performs all the necessary actions on one page.
The cons are that a Direct Post method isn't completely secure.

Client-side encryption

Client-side encryption, also known as encryption-at-source, refers to encrypting sensitive on the client-side device before sending it to the Merchant's server. This enables the Merchant to simplify your PCI compliance requirements. In a nutshell, it enables you to accept payments on your website while encrypting card data in your browser, using the payment gateway's encryption library.
Iframe
Provide your Merchant with a Javascript. The Merchant will embed it and set an iframe to open the payment gateway inside this iframe.

Hosted Checkout Button

Provide your Merchant with a Javascript. The Merchant will embed It, and define a button in his site, and the button will complete the payment on behalf of the Merchant.

2- The customer enters their credit or debit card details on the payment page. These details include the card-holder's name, card expiration date, and CVV number (Card Verification Value). This information is securely passed onto your payment gateway based on your integration (hosted payment page, server-to-server integration, or client-side encryption).

3- The payment gateway tokenizes or encrypts the card details and performs fraud checks before they send the card data to the acquiring bank.

4- The acquiring bank securely sends the information to the card schemes (Visa, Mastercard).

5- The card schemes perform another layer of fraud check and then send the payment data to the issuing bank.

6- The issuing bank, after performing fraud screening, authorizes the transaction. The approved or declined payment message is transferred back from the card schemes, then to the acquirer.

7- The acquiring bank sends the approval or decline message back to the payment gateway who then transmits the message to the Merchant. If the payment is approved, the acquirer

collects the payment amount from the issuing bank and holds the fund into your merchant account (more on that later on).

8- deposits the funds into the Merchant's account, a process which is known as the settlement; when the actual settlement will occur, it depends on the agreement the Merchant has with their payment gateway.

9- Based on the message, the Merchant may either display a payment confirmation page or ask the customer to provide another payment method.

Important features/factors in any Payment Gateways

Data Processing and Historical Payment Data

The simple reason being: That is what Payment Gateways are here to do, passing on information from one payment party to the other. It's not as simple as that, of course. Payment Gateways must be able to handle securely and process the data. Any leak might result in significant damage to reputation and finances.
Other crucial aspects of data processing would be reporting and management of historical payment data – important for recurring transactions, fraud prevention, callback handling and transaction reconciliation with PSPs.
Historical payment data is also where we enter the field of data analytics. Analytics functionalities give marketplace owners tools to optimize flows and costs based on preferred payment methods or to give out AI-calculated product or service suggestions to consumers.

Data Analytics and Reporting

Payment Orchestration Platforms greatly simplify payment data evaluation, due to its consolidating nature: All PSPs in a PO system are maintained at the same place. Compiling data reports for multiple PSPs at once is easily possible. This data can easily be shared as well, with merchants, fraud detection services, or financial authorities, for example.

Smart Routing

One of the major benefits of Payment Orchestration Platforms lies in their ability to dynamically route payments. And by setting up the routing rules, you gain greater control over the payment flows. For example, you could automatically funnel transactions through the channels that provide the best conditions. This could mean providing low-risk payment methods for high-risk customers or choosing a channel with low transaction costs.
Or it could imply processing the payment through a specific high-speed provider reducing latency. It's also possible to reduce the amount of failed transactions by re-routing them to

other acquirers or payment service providers, should the payment fail at the first one. This prevents frustration with your merchants and customers, too, which takes us straight to the next bullet point.

Multi-Party Transactions

As mentioned above, Payment Gateways used in online marketplaces will have to support transactions with more than two parties involved. The exchange of goods for money between the Consumer and the Merchant is expanded by the third party: The Marketplace Owner, who may charge a fee for each transaction or hold back money for the Merchant until specific conditions are met.

Additionally, customers of online marketplaces often bundle their orders and purchase from multiple merchants at once in a single basket/payment process. Payment Gateways have to unknot and properly assign and process such transactions, and real-world money flows, too.

Multi-Currency Support

The internet knows no borders. Of course, this also applies to all kinds of online commerce. Thus, multi-currency support is a must-have for marketplace owners. Consider choosing a Payment Gateway that can handle foreign currencies and authorize international payments. This does not only broaden the range of potential buyers. It also gives customers a feeling of familiarity. And this positively impacts the user experience – which takes us right to the next aspect.

User Experience

Cumbersome, non-transparent checkout and payment rank among the top reasons why consumers abandon their shopping cart mid-purchase. When planning how you want your marketplace's payment to play out, take on a consumer's mindset. They should never lose orientation during the checkout process on your marketplace platform. So make sure to present the relevant information when they are needed and keep the payment process simple and straightforward.
This might mean that you choose Payment Gateways, which don't redirect to other pages and adjust the UI for an optimal experience. Also, request only information from the customer, which is unavoidable for the transaction to proceed.

Merchant Experience

As said above, consumers are only one side of a transaction. Don't forget the merchants – they are users of your marketplace, too. For starters, make onboarding easy for merchants. Also, provide them with a clean UI, allowing them to report payment issues and take countermeasures.

Whatever you do, make sure to decide in favor of a Payment Gateway with recurring payment support. This way, your merchants can offer continuous services or subscriptions for products.

Fraud and Risk Prevention

It's not a question of having it or not having it: Your Payment Gateway must be secure. Fraud and risk estimation functionalities form the backbone of any Payment Gateway worth its code.
But it must not only be secure: It must also be perceived as secure by your customers as well as your merchants. Perception is just as important as reality. If a merchant or customer makes only a single experience with security breaches, fraud, or the like, they may stop using your marketplace platform.
Thus, you should pick a Payment Gateway with a high-security profile, including top-notch encryption standards, PCI DSS compliance, and tight KYC and Anti-Money Laundering processes. If you are dealing with high-volume purchases and large sums, you should not go without a Payment Gateway specialized in fraud detection.
Performance

If there is one thing no merchant likes, it's waiting for the money to arrive. While it's self-evident, that the online authorization by a Payment Gateway should only take seconds – don't use a slower Gateway unless you have good reasons, such as multi-level security calls – transferring the money from the customer's to the Merchant's bank account usually takes longer.
Mind though, that Payment Gateways work with what they come upon. Payment Service Providers and their associated Payment Processors play a huge role in how fast transactions will proceed. In any case, you should optimize the payment process as well as you can, so it runs smoothly. Ultimately, this means that you should have as much control over the outline and the functionalities of your Payment Gateway as possible.
Scalability and Compatibility

Payment Gateways are not built for eternity. Innovation in digital finance and online payment happens in a matter of months – take the rise of cryptocurrencies for example. Consequently, you may want to put extra focus on a flexible Payment Gateway, which you can easily adapt and update.
Ultimately, this equals building one yourself, as you cannot influence how third parties will update and maintain their Payment Gateway solutions. Yet, even if you settle for a turn-key product, you should at least aim for high scalability. Your Payment Gateway should be able to grow as your business grows and not suddenly hit the glass ceiling that is its own productivity.
Also, you must take into account that changes in the company providing the Payment Gateway for you might bring forth changes in the Payment Gateway's reliability.
Costs

While it's nice to dream about all the features you want your Payment Gateways to have, your budget tells you what is feasible. Compare costs and fees associated with different Payment Gateways and factor in development costs, if building your own Payment

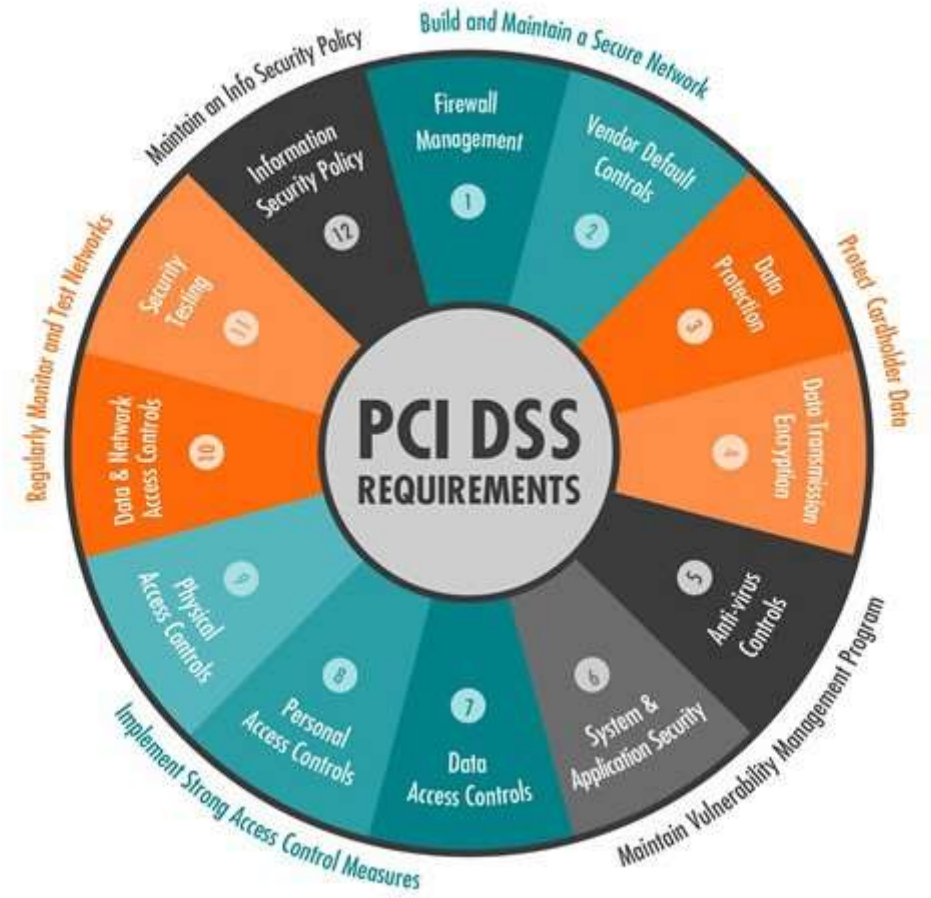
Gateway is an option. Payment Gateways that keep initial costs low, could make their profit via high fees and vice versa.

Important aspects to be taken in mind during the development of a payment gateway

Source: https://trimplement.com/blog/2020/06/payment-gateway-development-marketplaces/

What is PCI?

PCI DSS for Credit Card (CC) Processing, which stands for Payment Card Industry Data Security Standard that has 12 rules that enforce some level of security to protect Credit Card information but can be applied for any Personal Identifiable Information. In order to process credit cards, you may be subjected to PCI DSS Audit and certification, which may imply high costs or personal liabilities.



What PCI DSS compliance level do I need?

There are four levels of PCI DSS compliance. Deciding which one you need to meet is a complicated process but generally breaks down into four areas:

Collection: Will card-holder information be collected on the customer's browser, the Merchant's server, or the payment gateway server?

Storage: Will card data be stored on the Merchant's servers, or on the payment gateway's servers?

Transmission: How will card data be transmitted to the gateway?

Processing: Will card-holder information be processed by the Merchant or by the payment gateway?

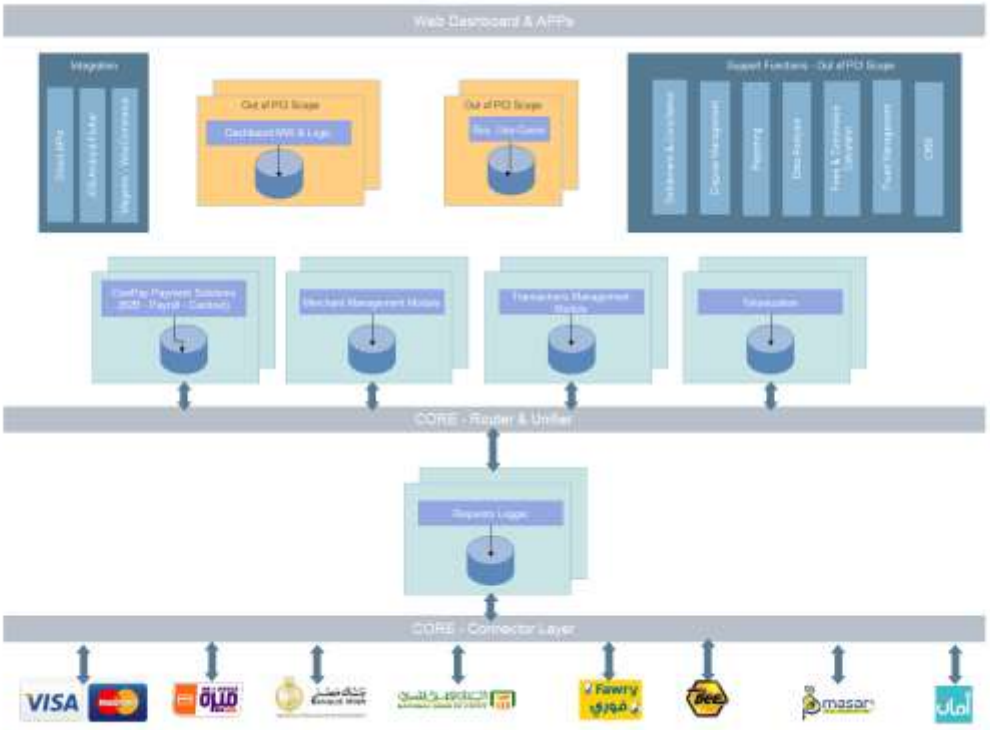
More information about PCI:
https://www.pcisecuritystandards.org/document_library

TECHNICAL SPECIFICATIONS

Our Final Target

- Building a Payment Gateway
- Campily with PCI
- Building Dashboard – reporting
- Building Fraud management
- Building White label solution
- Building an internal CRM
- Building a Gateways Management system
- Building a Dispute Management system
- Building Settlement and Conciliation Module
- Providing Payment methods:
    - Direct API integration
    - Hosted Checkout Integration
    - iOS – Android SDKs Integration
    - Flutter Plugin Integration
    - Magento, WooCommerc Plugins
    - Recurring
    - Installments
    - Bill Payments
- Providing Tokenization
- Offering Split out
- Providing the following method for cash withdrawal:
    - Mobile wallet
    - Bank account
    - Fawry Cashout
- Integrating with
    - MPGS
    - CyberSource
    - UPG
    - ValU
    - Bee for Bill Payment
    - Bee – Aman – Masary for POS
    - Fawry for Bill Payment
    - Fawry for installments
    - Fawry for POS
    - El-Bareed for POS
    - Fawry for Cash-Out

Main Architecture for the whole system



We are targeting implementing the different blocks as different services in sort of SOA implementation.

What we will care about in the current phase

Building minimal and basic payment-gateway features.
PCI certificate for the implemented features above.

What are the minimal features for the current phase?

1- Core Connector Layer
2- API-Requests-Logger
3- Smart Routing Module
4- Merchants Management Module
5- Transactions Management Module
6- Tokenization
7- Integration’s methods (APIs, SDKs, Plugins)
8- Queuing

Let’s dive into some details regarding what we have to do in the limited time plan:

1	Connecter Layer	Logger Manager	PCI compliance considerations
2	Gateway Management		
3	Merchant Management		
4	Fees & Commession Management		
5	Smart Router		
6	Transactions Management		
7	Settelment and Concialation Management		
8	Payment APIs + Payment methods (SDKs)		
9	Dashboard APIs		

1- Core Connector Layer:

The connector layer is responsible for receiving the request from the Routing-Layer, and the connector layer then sends it to the third party (either bank, PSP, or any other 3rd party). Then wrap the received response and pass it back to the Router-Layer

In the following structure

```
{
  "gatewayResponse": {}
}
```

For now, we are focusing on MPGS-Integration only.

For MPGS we want execute the following transaction types:

Authorization – a type of transaction used to check if a customer has enough funds to pay. It doesn’t include the actual money transfer. Instead, during authorization, a merchant ensures that a card-holder is capable of paying for an ordered item. An authorization transaction is used for orders that take time to ship/manufacture.

Capture – the actual processing of a previously authorized payment resulting in funds being sent to the Merchant’s account.

Sale – a combination of authorization and capture transactions. A card-holder is first authorized. Then funds may or may not be captured. It’s a regular payment for immediate purchases, like a subscription purchase, or e-tickets.

Refund – the result of a cancelled order for which a merchant will have to apply a refund payment processing to return the money.

Void – similar to refund but can be done if funds were not yet captured.

Tokenization – Gateway Tokenization allows you to store payment details in exchange for a token. The token replaces the payment details in the transaction request sent to the gateway. This is useful as the gateway handles the payment details collected from the payer thereby reducing your PCI compliance obligations. Further, if the token is stored with the payer data, it may be used when the payer returns to make another purchase.

A token is an identifier of stored payment details; returned when you tokenize them. The token may be used for all subsequent payment transactions to refer to the previously saved payment details.

Tokens are in PAN format and will pass simple card validation rules, so that they can be stored in place of credit card numbers. However, their generation is designed to minimize the likelihood that they will be valid card numbers.

Key Benefits of tokenization

- Reduces PCI compliance costs as you do not handle or store any payment details.
- Reduces internal fraud as your staff has limited access to payment details.
- Allows you to update payment details stored against a token. This is useful when payment details expire/change or the payer wishes to change the payment method.
- Facilitates ease of integration of tokens into systems that currently expect card numbers. Tokens generated by the system can appear like card numbers and pass basic card validation checks.
- Allows you to retrieve payment details from a token.
- Offers different options for verifying payment details.
- Provides flexible options for token management.
- Allows you to share tokens with other merchants.

Examples:

- Recurring Billing (utility bills, gym membership, etc): Collect payment details from the payer and store them in exchange for a token. The token is submitted to the gateway as the payment instrument each time a payment is due. This is useful if you wish to reduce PCI costs.

- Online Retailers (online shopping carts, online bill payment, gaming sites): Collect payment details from the payer on a website and store them in exchange for a token. The returned token is stored with the payer data. When a payer returns to the website to make another purchase, you present the masked account identifier (or last four digits of the token, if using the Preserve 6.4. Generation Strategy) and indicate that payment details do not have to be re-entered. This saves payers from having to re-enter some or all of the payment details. It improves convenience and payers' user experience when making purchases through your website.

Sale using a token – Same as sale but by using token instead of Card-date

Recurring Payment – Three types:

- Regular subscriptions
- Unscheduled payments (like pay as you go approach)
- Instalments

To implement these operations, we need to call some rest-APIs from MPGS-APIs:

https://banquemisr.gateway.mastercard.com/api/documentation/apiDocumentation/rest-json/version/latest/api.html?locale=en_US

- Authentication: Authenticate Payer
- Authentication: Initiate Authentication
- Gateway: Check Gateway
- Gateway: Payment Options Inquiry
- Session: Create Checkout Session
- Session: Create Session
- Session: Retrieve Session
- Session: Update Session
- Tokenization: Create or Update Browser Payment Token
- Tokenization: Create or Update Token
- Tokenization: Create or Update Token (with system-generated token)
- Tokenization: Delete Token
- Tokenization: Retrieve Token
- Tokenization: Search Tokens
- Transaction: Authorize
- Transaction: Balance Inquiry
- Transaction: Capture
- Transaction: Pay
- Transaction: Referral
- Transaction: Refund
- Transaction: Retrieve Order
- Transaction: Retrieve Transaction
- Transaction: Update Authorization
- Transaction: Verify
- Transaction: Void

We want to implement all of the calls by covering both mandatory and optional fields to be ready for any integration type using MPGs in the future.
In MPGs to perform the main operation (like pay, void, refund, etc.) you have to follow a workflow of the mentioned APIs.

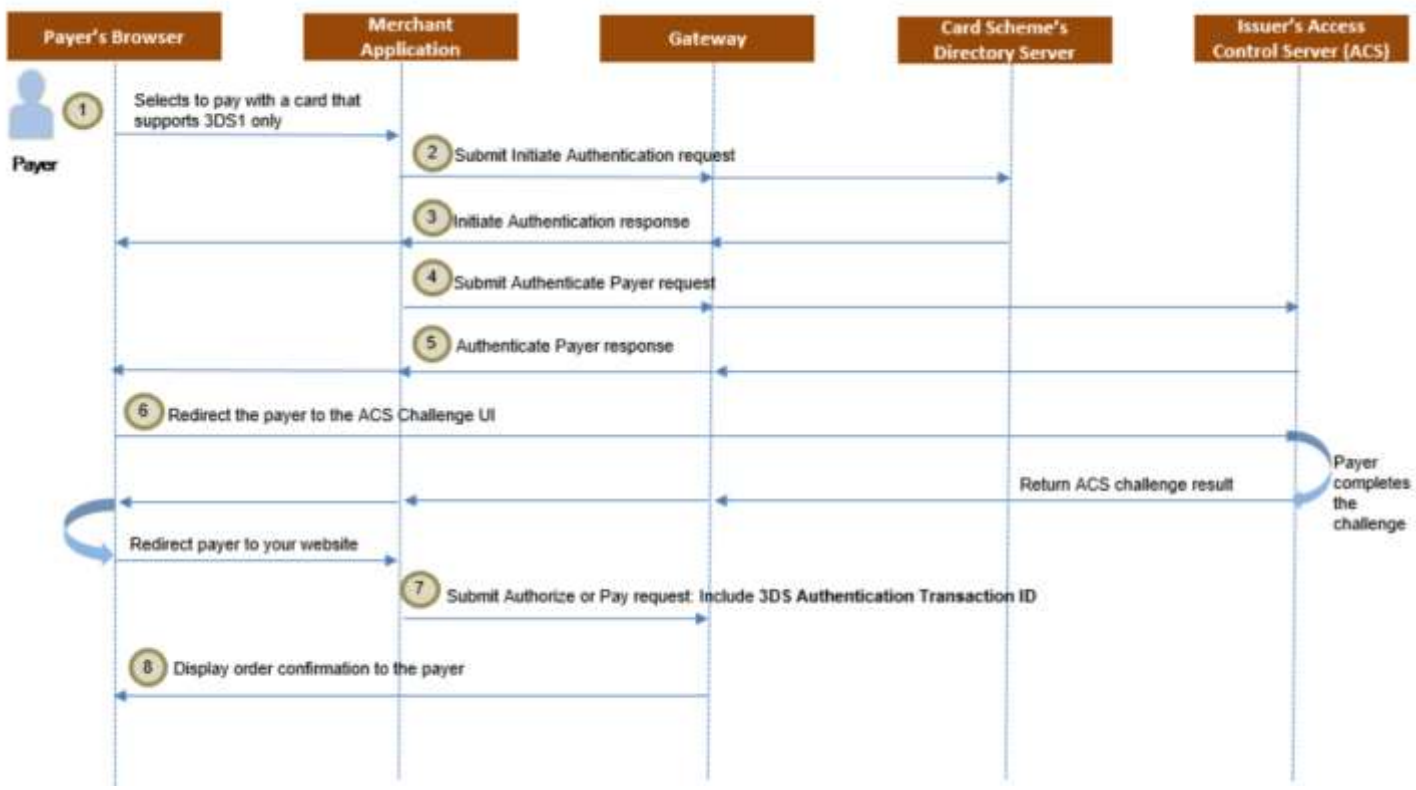
Postman collections will be attached to demonstrate the basic scenarios with mandatory fields. But we will need to develop the layer with all of the optional fields as mentioned.

Also there is a WADL definition here:
https://banquemisr.gateway.mastercard.com/api/documentation/apiDocumentation/rest-json/version/latest/wadl.xml?locale=en_US

The router-layer will pass the required MPGS fields for each series of calls.

Important information and documentation from MPGS docs

3DS (3 Domains Secure)
https://banquemisr.gateway.mastercard.com/api/documentation/integrationGuidelines/supportedFeatures/pickAdditionalFunctionality/authentication/3DS/3DSecureAuthentication.html?locale=en_US



Implementing a 3DS Integration using the Authentication API

https://banquemisr.gateway.mastercard.com/api/documentation/integrationGuidelines/supportedFeatures/pickAdditionalFunctionality/authentication/3DS/integrationModelAPI.html?locale=en_US

Aggregator Support
https://banquemisr.gateway.mastercard.com/api/documentation/integrationGuidelines/supportedFeatures/aggregatorSupport.html?locale=en_US

2- API-Requests-Logger:

This Service will provide an API to log all of the APIs requests across all of our layers (services).

taking in consideration the following:

1- Some logs will be used with the transactions logs in dispute system.
2- Some logs will be used in debugging failed transactions.
3- Some logs will be used to debug failed communications between different layers (services).

So, this layer (service) will be used commonly for different purposes.

NoSQL Database can be suggested to fit here.

3- Smart Routing Module

It is a layer (service) that conveniently sits inside the payment gateways to assist the payment processing.

Being an OLTP (online transaction process tasks) it takes care of all the nuances inside a transaction. Suppose the payment gateway is a founder who oversees the transaction. In that case, the router is the executive who does the actual payment processing tasks.
In a payment gateway, there are multiple merchant accounts boarded with their bank (acquirer bank). So, when a payment request comes from a merchant selling platform, router dynamically identifies the acquirer bank (associated with that Merchant) and the issuing bank of that specific request through BIN allotment and then permits the transaction to happen securely.

BIN allotment is one among many ways for the payment router to route the transaction. It also supports routing by amount, routing by the time of the day. Once it receives the message from the issuing bank, it formats and sends back the response to the acquirer. The router is a flexible entity accepting the payment request from the payment gateway and beginning the transaction steps.

- Accepts the validated payment request from the PG.
- Reads the merchant rules for the transaction process.
- Identifies the PSP for that particular payment request.
- Routes (switches) the transaction based on a BIN allotment for a specific PSP associated with that payment request.
- Processes the transaction based on failure or success.

4- Merchants Management Module

A service to manage all of the merchants (clients) information, with its own separate database. It will maintain:

- Merchants' info (Bank account, legal papers, etc.)
- Merchants' Fees and Rates
- Merchants' MIDs
- Merchants' Configurations
- Handle Merchant Teams
- It will interact with the transactions layer/service to get:
- Merchants' transaction history
- Merchants' Volume
- Merchants' Balance
- Handle Withdrawals's requests (with transaction layer/service)

Transactional ACID database is recommended.

5- Transactions Management Module

A service to maintain all of high-level transactions. For example, Payment transaction will be saved as a one record in the database (regardless all of its multiple MPGS Calls)
It will log all of the high level operations:

1- Payment
2- Refund
3- Charge Back
4- Void
5- Auth
6- Capture
7- Partial Capture
8- Partial Refund
9- Withdraw request

Transactional ACID database is recommended.

6- Tokenization

Tokenization, the process of protecting sensitive data by replacing it with a token, is often used to prevent credit card fraud. In credit card tokenization, the card-holder's primary account number is replaced with the token. The token is then passed through the various networks needed to process the payment, but actual bank details are never exposed because they are held in a secure token vault.
Tokenization in and of itself won't make a merchant PCI compliant, but it is considered a "best practice." It can help reduce PCI DSS scope.

Credit card tokenization concept

The idea behind tokenization is to delegate credit card storage to the payment gateway, as opposed to business-specific software, such as website, online storage or CRM. The main purpose of tokenization is to allow merchants not to store credit card numbers for repeated\recurring transactions. To achieve this, credit card numbers are replaced by tokens which are saved and used instead.

Fundamentally, there are two primary approaches to tokenization.

Approaches to credit card tokenization as a process

The two conceptual ways to implement credit card tokenization are commonly referred to as pure tokenization and profiling.
Pure tokenization
Under pure tokenization approach, only credit card (bank account) number is tokenized. However, if necessary, routing number (identifying the branch of the bank) for ACH, ID number, or even driver license number, could be tokenized as well. To put it simply, every sensitive value is replaced by a token (one for one).

When the credit card processing request is issued (during repeat\recurring transactions), the token is used instead of a credit card number, allowing merchants to avoid credit card storage.

Profiling

The second approach to credit card tokenization (profiling) is a bit more elaborate, as it involves maintaining the full or partial customer profile.

The difference is that in this case the entire billing information, including the billing address, shipping address etc (depending on the system used) would be stored in the customer profile (including credit card number). When a transaction is processed, the ID of the profile is sent, and any fields that are not supplied, are "pulled" from that customer profile.

Some systems use a variation of customer profiles where instead of customer profile an ID of the previous transaction is passed and all of the missing information, such as credit card number etc, is extracted from that transaction.

When recurring credit card transaction is processed, the profile ID (or ID of the previous transaction) is used in place of the actual credit card number, thereby allowing the Merchant to avoid persistence of the credit card number.

Advantage of the first approach is that the business doesn't have to do any separate maintenance of the profile. It only has one number and the token just replaces it. Advantage of the second approach is that more information can be stored outside of the Merchant's system (web-site\CRM). Consequently, if the Merchant has some basic front-end system and doesn't wish to store all the billing information (ZIP code etc), it can rely on tokenization provider to store that information, which might be more convenient for the Merchant. However, with this approach the Merchant assumes the responsibility for keeping the profile up to date.

Now let us look at the most common ways to implement credit card tokenization.
Credit card tokenization through appliance

Tokenization through appliance is intended for the pure tokenization approach, described above.

The appliance is a combination of a hardware device (used to encrypt\decrypt credit card numbers) and PA-DSS-compliant software (used to store encrypted values and to generate tokens). The hardware device is usually either a chip on the motherboard or a PCI-card. The appliance (hardware with software written on top of it) is hosted in local network of the merchant\PSP.

Hardware\appliance solution does not eliminate the need for PCI-compliance certification, but it definitely reduces the scope and simplifies the PCI audit, as the storage is delegated to an already PA-DSS-compliant piece of software.

Credit card tokenization through appliance can be an ideal solution for a merchants\PSPs processing large volumes of transactions and already having an existing PCI environment where the appliance could be deployed.

More information can be found here:
https://banquemisr.gateway.mastercard.com/api/documentation/integrationGuidelines/supportedFeatures/pickAdditionalFunctionality/tokenization/tokenization.html?locale=en_US#x_tokenConfiguration

7- Integrations methods (APIs, SDKs, Plugins)

We target providing the following to our clients:

- Direct API integration
- Hosted Checkout Integration
- iFrame
- iOS – Android SDKs Integration
- Flutter Plugin Integration
- Magento, WooCommerc Plugins
- Recurring
- Installments
- Bill Payments
- Tkoneization
- Split out

8- Queuing

Utilize queues to make your system more reliable and resistant to change and load. As part of this step, you can create a software design where all communication, or most of it, between components is done via queues (where async is permitted). For example, you may have queues for internal Fraud scoring, external Fraud scoring (aka MaxMind or ThreadMetrics), storage, notifications of other systems, a queue for every Payment Processor, and many more moving pieces.

APPENDIX A
GLOSSARY

3D Secure

An authentication method used by merchants to validate cardholders. The cardholder authenticates their card against the Issuing Bank's website. The merchant chooses whether or not to use 3D secure, and this is usually done via an iframe on the merchant's site. This allows the merchant to shift liability from themselves to the issuing bank in some cases. 3D Secure requires cardholder interaction to be completed.

ACH

Automate Clearing House, an electronic network for financial transactions. An alternate payment method that uses the routing and account number for US bank account holders to electronically transfer money to the merchant.

Acquirer (Acquirer Bank)

An acquiring bank is a bank or financial institution that processes credit or debit card payments on behalf of a merchant. The acquirer will either approve or decline a credit card transaction. If approved the acquirer will then settle the transaction by placing the funds into the merchant's account. The term acquirer indicates that the bank accepts or acquires credit card payments from the card-issuing banks within an association.

Acquirer BIN

This number (typically a 6 digit number) represents the bank identification number within an association. So, if Chase Paymentech is an acquirer, it will have a different bin for Visa than Mastercard. The acquirer bin is used for enabling 3D Secure.

Action

Part of the Rules Engine relating to a PHP class name to be called when certain events trigger a rule and rule conditions match.

API

How merchants can interact with Rebilly.

API Authentication

How API calls are authenticated.

API Key

The API secret or publishable key. A user can have multiple keys. A user can be a member of one or more merchants. And a merchant can have multiple members.

Apply

A coupon, once redeemed, will be applied to the customer's invoice automatically if it fits the coupon's restrictions.

Approval

A result of a transaction that is considered valid and accepted by the Issuing Bank.

Audit

A compliance or financial evaluation that takes place once or more per calendar year.

Auth (authorization)

A transaction type used to authorize that a card is valid, and results in a hold placed on the cardholder's account for the authorized amount. An authorization response code is later used to capture the authorized funds.

AVS

Address Verification System. Used in fraud prevention on credit and debit card transaction by verifying a cardholder's billing address.

Balance

The amount a customer owes the merchant. The balance is calculated by adding up all debit invoice items and subtracting credit invoice items (on invoices that aren't abandoned or voided), and then also subtracting all approved sale/capture transactions, and adding approved refund/void transactions.

Billing

The general process of sending payment requests and receiving payments.

Billing Address

The address where the issuing bank mails the cardholder's monthly statements. Also the address an invoice should be addressed to -- which may be distinguished from the delivery address (the address where products may be delivered to.)

BIN (Issuer Bin)

The first 6 digits of the payment card which typically designate the issuing bank. An issuing bank may have multiple bins.

Bitcoin

A cryptocurrency, and a potential alternative payment method.

Blocklist

A stored value that if matched results in aborting an operation. Rebilly has blocklists of customer Ids, emails, ip addresses, bins and payment cards. A blocklist can be not permanent, by setting up a TTL (time to live) as an expiration date.

Cancellation

The stoppage of a customer's subscription.

Capture

The action describing capturing payment card funds that are authorized. An authorization plus a capture is similar to a sale. A capture debits the funds from the cardholder's account, and on settlement will be moved to the merchant's account.

Category Code

Merchant category code. A merchant category code (MCC) is a four-digit number assigned to a business by credit card companies (for instance American Express, MasterCard, VISA) when the business first starts accepting one of these cards as a form of payment. The MCC is used to classify the business by the type of goods or services it provides.

Chargeback

A cardholder can dispute a charge with their issuing bank. If the issuing bank accepts the dispute, it becomes a chargeback. A chargeback is sent through the card association to the acquirer, eventually on to the merchant. The merchant has a deadline to respond to the chargeback. Responding is typically called "representment." A chargeback has a reason code (the reason codes are specific to and predefined by each of the card brands; such as Visa, MasterCard, American Express, etc.)

Check

A bill of exchange where the drawee is the bank. A check, in the US, has a routing number and an account number. It's written for a specific amount, to a specific payee (or to cash). A check is good for only a single payment.

Checkout

Hosted forms for merchants for creating a subscription for or collecting payment from a customer. Interaction with the form is by the customer.

Compliance

Ensures a body (i.e. - Rebilly) is aware and taking steps to comply with issued regulatory laws, rules or specifications set for an industry.

Consecutive Decline

The number of times a payment card was consecutively declined while authorizing an auth or capture transaction.

Coupon

Coupons allow you to apply different types of discounts to invoices, subscriptions and plans.

Credit

In accrual accounting, a credit is the opposite of a debit. A credit is a positive amount added to an account.

Credit Card

Sometimes used erroneously to mean all payment cards, but it is different from a debit card in that a line of credit is extended to the consumers. More specifically, a consumer is not expected to have funds available for a credit card purchase, and transactions within their credit limit are financed with the expectation of future settlement with the issuing bank.

Currency

A system of money in common use especially in a nation. For example, US dollar, British pound, etc.

Customer

A customer is an entity that is purchasing goods or services from a merchant, and will be the payee in any transaction that is credited to the merchant.

- customers are associated with payment cards, subscriptions, invoices and other miscellaneous relationship models.

CVV (cvv2/cvc/cvc2)

A code found on and associated to a payment card used for verification purposes.

DBA

Doing Business As. A merchant may be doing business under another name.

Debit

In accrual accounting, a debit is the opposite of a credit. A debit is an amount that is subtracted from an account.

Debit Card

A payment card associated with a bank account that allows a client to pay for goods and services electronically. A debit card is different from a credit card in at least that the funds are automatically debited from the cardholder’s bank account at time of payment authorization.

Decline

A response where the issuing bank declines to approve a payment transaction. A decline can be for various reasons, and it will accompanied with a reason code, such as: insufficient funds, incorrect billing address or cvv, etc.

Delivery Address

The physical address where goods are delivered to a customer. May be the same or different than the billing address.

Direct Debit

A payment system whereby creditors are authorized to debit a customer's bank account directly at regular intervals.

Discount Rate

The percentage of fees that the processor charges the merchant in addition to any other fees the processor may charge the merchant.

Dispute

A Rebilly term that collectively means chargebacks and retrievals.

Dispute Deadline Time

The deadline that a merchant must submit a representment by for a dispute. If the deadline is missed, the merchant loses the dispute.

Dispute Reason Code

The code used in a chargeback that reflects the reason for the dispute.

Dunning

Dunning is the process of methodically communicating with customers to ensure the collection of accounts receivable. In Rebilly's terms and use, it is the process of retrying unsuccessful payment transactions.

Dunning Discount

A discount that a merchant may wish to grant the customer in exchange for debt collection of a due balance.

Dunning Index

The order in which dunning schedules are applied to the dunning process.

Dynamic Currency Conversion

Dynamic currency conversion (DCC) is the feature to present to the customer the choice to be billed in the original transaction currency or the customer's native currency.

Dynamic Descriptor

A descriptor is the information that appears in a cardholder's billing statement to clearly identify the source of a credit or debit card transaction. A Dynamic Descriptor is a descriptor that can be modified on a per transaction basis.

Exchange Rate

The rate used in the transferring of one currency to another.

Exchange Time

The time a currency exchange occurred that is used to lock in an exchange rate for the exchange.

Event Types

In the Rules Engine, a rule set that is configured will be executed after a certain event was fired.

Forecast

Predicting future revenues, cost and other behavior and metrics based on historical data and other factors.

Fraud

When a customer uses deception to receive financial gain.

Gateway

A payment gateway is an e-commerce application service provider service that authorizes credit card payments for e-businesses, online retailers, bricks and clicks, or traditional brick and mortar. It is the equivalent of a physical point of sale terminal located in most retail outlets.

Gateway Credentials

The credentials given to a merchant by the merchant's payment gateway/processor.

Grids

Tables of data able to be viewed in the Rebilly app by a user. Grids are customizable as users are able to add and remove columns and edit the column order.

Installment Payments (hire purchase)

The legal term for a contract in which a purchaser agrees to pay for goods in parts or a percentage over a number of months. Other analogous practices are described as closed-end leasing or rent to own.

Invoice

An invoice is a commercial document issued by a seller (merchant) to a buyer (customer), relating to a sale transaction and indicating the products, quantities, and agreed prices for products or services the seller has provided the buyer. Payment terms are usually stated on the invoice.

- Invoices are tied to a Customer, Currency and Website
- Invoices have an issuedTime, dueTime and closedTime (paid).
- Invoices may also have a voidedTime or abandonedTime (if voided or abandoned).

Invoice Item

Refers to a line on the invoice, which may contain information about the product, quantities, duration and prices.

Issuing Bank

The bank that issued the payment card to the cardholder.

Late Fees

Fees a customer may be charged, in addition to the balance due, if their payment is not received by a specific due date. Some merchants allow a grace period before charging a late fee.

Layout

A collection of plans, in a specific order, which a merchant may present to a customer (or prospective customer) on a “pricing” or “plans” page.

Like for like

A phrase that means processing and settling in the same currency.

Live vs Sandbox

Sandbox is an isolated testing environment that allows the testing of configurations and code via the merchant before setting themselves live. A merchant can use the sandbox after going live, as well.

MCC

Merchant category code. See Category Code for full definition.

Merchant

A merchant sells a product to end-users or consumers.

MIDs

A merchant account identifier.

One-time Charge

An ad hoc amount that a merchant can charge a customer via the Rebilly API. This amount is not expected or scheduled to recur in Rebilly’s system.

PAN

Payment account number (also known as the credit card number of debit card number) -- typically 12-16 digits long. The PAN must be protected per PCI Compliance.

Partial Payment

Different from an installment payment, a partial payment represents a customer who has paid any amount less than the balance due. The merchant may be expecting a partial payment, or may not be expecting it.

Past Due

A customer balance greater than 0 that is past the due date of an invoice.

Payment

The action or process of paying for goods or services. A payment creates a transaction.

Payment Collection

The process of collecting the balance due from a customer on behalf of a merchant.

Payment Method

The type of payment used to settle a balance due. Some examples are:

- payment card (credit or debit)
- check
- wire
- ACH
- direct debit
- etc.

PayPal

Both an acquirer and an alternate payment method.

Paysafe Card

An alternate payment method that is a prepaid online payment card.

PCI

Stands for Payment Card Industry.

PCI Compliance

PCI (Payment Card Industry) has a set of regulations which Rebilly must comply with.

Plan

A template of the key terms describing a subscription. A plan may have optional setup fees, an optional trial period and fees, and an optional recurring fees and frequency. A plan may also expire, or be valid only for a limited number of recurrences.

Prepaid Card

A payment card that has funds loaded to an account prior to it being used, and will work just like a credit card. Some prepaid cards are reloadable, and some are not (those are disposable).

Pricing

The amounts set up for specific plans that a merchant may offer.

Processing Limits

A cap (limit) of the amount that a processor is willing to process for the merchant for some frequency. In Rebilly, we use the calendar month as the period frequency.

Processor

Also used interchangeably with Gateway, means to describe the mechanism that provides communication transport to the Acquirer or Acquirer's platform.

Product

The product a merchant is selling. A merchant may sell many products. A product may be attached to a plan.

Product Bundle

Some merchants wish to bundle products (in effect making a super-product.)

Reconciliation

The process of checking merchant account statements (from merchant's processor/acquirer) against actual deposits (in corporate bank account) against sales records (in Rebilly.)

Rebill

The slang term for a recurring payment.

Rebill Number

The number of recurring payments in a subscription excluding trials or setup fees.

Rebilly App

Our merchant-facing web application.

Recurring Amount

The agreed balance that a customer will pay in recurring intervals for a subscription.

Redeem

To redeem is to attach a coupon to customer.

Refund

A refund removes funds from the merchant and transfers them back to a customer. A refund is similar to a void, except a void must happen before settlement of funds.

Renewal

A customer agreement to continue a subscription for a new period of time.

Report Filters (aka - extended filters)

The ability to filter report data by a variety of fields. The fields may or may not appear on the actual report.

Representment

The term used for the merchant's reply to a chargeback, with the intent to prevent the Acquirer from taking back the funds previously settled.

Reserve

The amount of funds a processor withholds from a merchant, usually in a specially designated reserve account. The purpose of the reserve is risk mitigation for the processor. The reserve amount may be nothing, a flat amount, or a percentage of transactions. The reserve may also be a rolling reserve, in that the amounts in the reserve are released after some period of time. A typical period of time for a rolling reserve may be six months.

Response

This term represents the response message, typically meant as the reason for a decline, from a gateway. Each gateway has unique response messages. Some gateways may also have response codes.

- Response Type
- Approved
- Declined Soft - means the transaction was declined, but may work at some point in the future.
- Declined Soft Fix - means with additional data (or data correction) from the customer, a transaction may be retried.
- Declined Hard - means the transaction is not likely to work, it does not need to be retried. An example is “No such card issuer,” and - “Transaction not allowed,” “Stolen card."

Restrictions

Settings to be applied to a coupon that restricts the coupon's use to a specific set that the user controls.

Retention

The continued use or payment of a good or service by a customer.

Retrieval (aka information request)

The process where an issuer requests information about a transaction through the card association to the acquirer, which passes it onwards to the merchant. A retrieval usually precedes a chargeback.

Rule

An event with a trigger, which evaluates conditions to allow merchants to customize parts of the application logic to execute configurable actions, which helps them meet their specific goals. Some logic that can be tailored via the rules engine is:

- processor selection
- blocklisting
- payment retries
- etc.

Sale

A transaction type that captures funds from a customer. A sale can be voided prior to settlement, and it can be refunded after settlement. A sale debits a customer account and credits a merchant account.

Schedule

The plan or configuration to carry out a subscription. Elements include:

- duration - the length of time a plan is valid till from start.
- time unit - the unit of time used to define a plan’s interval, such as: day, week, month, bi-monthly, quarterly, semi-annual, annual, etc. number of units - the number of time units that occur between each plan interval
- repeats on - the day of week, the day of month or the anniversary that an interval repeats on.
- number of occurrences - how many intervals will be used in a plan. A plan with no end is defined with 0 number of occurrences.

Setup Amount

A fee charged to a customer for setting up a new subscription. Usually a one-time charge.

Status

The state of a model entity, such as pending, active, inactive, etc.

Stop

The stopping, possibly due to cancellation, expiration or other reason(s), of a subscription.

Stop Reason

The reason a subscription stopped.

Subscription

A subscription is an instance of a plan for a specific customer and website combination.

A subscription has some properties:

- has a plan
- has a customer

- has a website
- has an external Id
- has a start time
- it may have an end time, cancelled time, and renewal time.
- it has a rebillNumber, which increments each time it renews.
- it has a quantity
- it has a product
- it has a billing address, and may have a delivery address

Tailored Pricing

The ability to create unique prices specific to a customer or group of customers for specific products or product bundle. A merchant may wish to charge two customers different prices for the same exact product. This pricing strategy is very common in SaaS models.

Tokens

A one-time use string that represents a customer's payment card details. A token expires within 24 hours.

Trial

A trial is an attribute of a Plan (model) that indicates that there is a period that has different pricing, with the possible inclusion of being free.

Transaction

An instance of an action regarding a payment.

- a transaction has a result (approved, declined, canceled, abandoned or unknown).
- a transaction has an amount and a currency.
- a transaction is related to a specific merchant website.
- a transaction is always related to a customer.
- a transaction can be related to another transaction, such as a void of a previous sale.
- a transaction is related to a payment method.

- a transaction cannot be deleted, and the only property of a transaction that can be updated is the result.

Transaction Types

- authorize - payment authorization
- capture - capture of funds from a previous authorization
- credit - transfer of funds to a customer that may not reference a previous transaction
- refund - returning of funds from a previous settled transaction
- sale - similar to an auth and capture
- void - cancelling of an unsettled capture or an authorization more transaction types may be added with new payment methods

User

An entity that can be authenticated and granted access to the Rebilly system either via API or app.

User Roles

A defining attribute of a user that determines their level of access and/or privileges in the system.

Void

A transaction type that reverses an authorization or capture/sale transaction. A void can only occur before settlement, otherwise a refund must be used for a merchant to credit a customer.

Webhooks

An HTTP request to a merchant defined url notifying the merchant in a programmatic way of certain events, such as: a new transaction, a new subscription, a new invoice, etc.

Website

Similar to a store, a website is a place that a customer is linked to merchant via a subscription. A website typically determines the processor account and billing descriptor used for payment transactions.

Weights

A method of attaching a priority or value to an entity that is being programmatically selected amongst others. Used in processor selection, and plan layout selection. The higher the weight, the more likely it is selected.

Wire

An alternate payment method. An electronic transfer of funds. Wire transfers typically transfer money from one bank account to another.