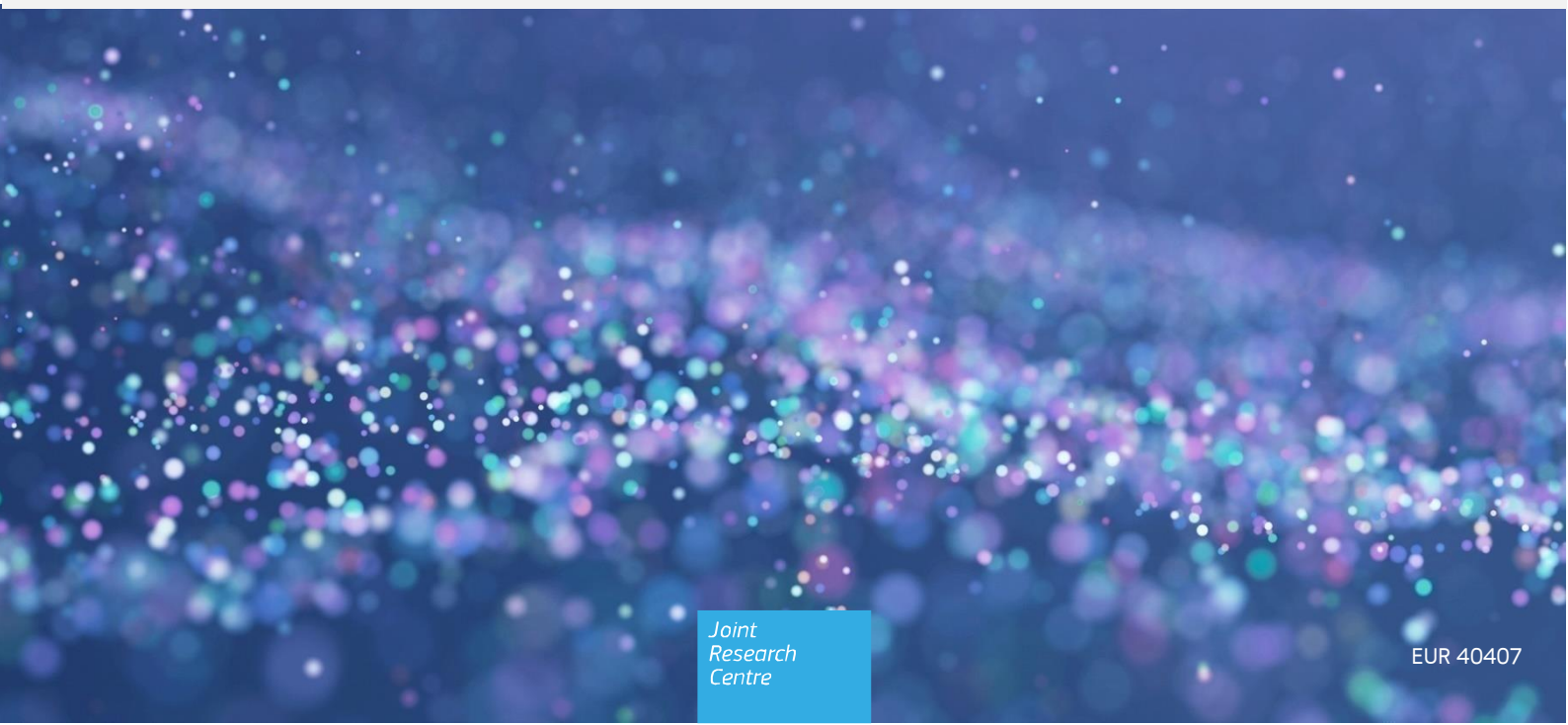
ISSN 1831-9424

Study on online payment ecosystem

Christaras, V., Karopoulos, G., Kounelis, I., Krasovec, A., Nai Fovino, I., Spigolon, R., Sportiello, L., Steri, G., Trapani, M.

2025



Joint
Research
Centre

EUR 40407

EU Science Hub
https://joint-research-centre.ec.europa.eu

JRC143177

EUR 40407

PDF ISBN 978-92-68-30945-2 ISSN 1831-9424 doi:10.2760/8195462 KJ-01-25-404-EN-N

Luxembourg: Publications Office of the European Union, 2025

How to cite this report: Christaras, V., Karopoulos, G., Kounelis, I., Krasovec, A., Nai Fovino, I. et al., *Study on online payment ecosystem,* Publications Office of the European Union, Luxembourg, 2025, https://data.europa.eu/doi/10.2760/8195462, JRC143177.

Contents

Abstract

This document is a study performed by the JRC to support an analysis of DG-Competition concerning online payment systems. The report provides a high-level overview of online payments in the EU, including payment cards, mobile wallets, bank transfers, and other methods, as well as an examination of APIs and SDKs used by online payment platforms, such as Samsung Pay, Google Pay, Amazon Pay, Blik, Paypal and Vipps MobilePay. Additionally, it discusses the use of blockchain technology in online payments, including stablecoins, credit/debit cards backed by cryptoassets, and crypto payment gateways. The report aims to provide a comprehensive understanding of the online payment landscape in the EU, highlighting its various components, technologies, and innovations.

1. Introduction

This document is a study performed by the JRC to support an analysis of DG-Competition concerning online payment systems.

Following a specific request of DG-Comp, it is organised in three main sections:

— **Online payments in the EU**, providing a high-level overview of mobile wallets, payment cards, bank transfers, BNPL, instant payments, payment initiation services, cash on delivery;

— **APIs/SDKs for online payments:** it examines online payment methods' APIs and SDKs on iOS and Android devices, focusing on Google Pay, Samsung Pay, Amazon Pay, PayPal, Blik, and MobilePay. It also, where possible, describes the related architectures and compare APIs accessible to third parties;

— **Blockchain technologies in online payments:** it provides a high-level overview of use-cases where blockchain based technologies are used in online payments.

2. Online payments in the EU

EU citizens have diverse possibilities when they intend to operate online payments. Here below we try to summarize them, highlighting their main features and the way they can be accessed by the users.

2.1. Payment Cards

Payment cards are widely used for online purchases and are one of the historical means of payment in such a context. They are offered in the form of plastic cards with data embossed on its surface, like card number and holder name. They operate on global card networks (like Visa and Mastercard) and can be of different types: credit cards allow to borrow money which has to be repaid by a due date typically charging the amount on a linked bank account, debit cards are linked to a bank account and allow to spend money directly from the bank deposit, prepaid cards require that money is loaded onto the card before using it and thus limiting the spending to the amount loaded onto the card. During online payment checkout, users enter the card details, including the card number, expiration date, and usually the card security code (e.g., CVV/CVC). While convenient, card payments may present some additional risks compared to other payment methods, due to the communication of card sensitive data and their possible storage by the recipient.

2.2. Mobile Wallets

Mobile wallets, also known as digital wallets, enable users to securely store payment information on their smartphones. They are usually offered in the form of applications on the mobile device and allow for convenient and contactless payments in various scenarios — whether at physical stores interacting with a payment terminal, but indeed also during online shopping or within mobile apps. Users typically have to physically act on their phone to complete transactions at the moment of online payment checkouts, like for instance inserting a PIN, performing a biometric identification or scanning a QR code, actions basically adopted for confirmation of the payment. To enhance security, mobile wallets can rely on hardware secure elements to store payment information or use a process called tokenization, which substitutes sensitive data (such as credit card numbers) with tokens during payment processing. Additionally, some mobile wallets offer the feature of storing their own credit. Users can preload funds into the wallet, which effectively operates like a virtual prepaid card. This balance can be used for future purchases without directly charging a linked credit card or bank account.

2.3. Bank Transfers

Bank transfers is an electronic method to move money directly from one bank account to another, and it is one of the most traditional ways of making payments. To initiate a transfer within the bank systems it is necessary to provide essential details about the recipient, typically name of the account holder, account number and purpose of payment. The recipient's bank then digitally receives the payment. Customers can initiate transfers manually (account-to-account) throughout the bank services like a banking app or the online banking platform, or through third-party providers leveraging the Open Banking scheme (see Payment Initiation Services below). Specifically for the EU the SEPA Credit Transfer (SCT) scheme is available, it allows to make bank transfers within the Single Euro Payments Area (SEPA) which includes 36 European countries (the 27 EU members and other countries participating to the scheme). The process relies mainly on the International Bank

Account Number (IBAN), which is a standardized system for identifying bank accounts across countries, with transactions that are typically completed within one business day.

2.4. Instant Payments

Instant payment systems enable real-time fund transfers among bank accounts, and they are based on transactions that settle within seconds of initiation. Unlike traditional payment methods such as bank transfers, which may take hours or days to clear, instant payments provide immediate funds transfer. In the EU they are available as SEPA Instant Credit Transfer (SCT Inst) scheme. These systems offer 24/7 availability, are capped at 100000€ and allow funds to be made available on the recipient's account in less than ten seconds and therefore allow to get immediate confirmation of a payment. Users can typically launch an instant payment using a banking app or an online banking platform.

2.5. Payment Initiation Services (PIS)

Payment Initiation Services (PIS) are a type of financial service enabled by the Open Banking scheme. This concept has been introduced with the EU's PSD2 (Payment Services Directive 2), which mandates banks to open their data to third parties with the customer's consent, so allowing to access banking information like transaction history and account balances. These services allow consumers to initiate payments from their bank accounts without needing to access their bank's online banking interface. Indeed, when making a purchase online a consumer, instead of using a credit card or other payment methods, selects the PIS option at the checkout. The PIS provider connects to the consumer's bank account, with the consumer authorizing the payment typically using a form of strong customer authentication as indicated by the PSD2.

2.6. Buy Now Pay Later (BNPL)

Buy Now Pay Later (BNPL) is a financial service that allows consumers to shop online, receive goods, and pay for them over time, usually in instalments, typically without incurring interest or fees if payments are made on time. With BNPL the consumer selects the BNPL option at the checkout, the BNPL provider performs a quick credit check before approving the transaction and paying on behalf of the consumer. The total purchase amount is divided into small instalments paid by the consumer over a specified period of time. For such payments the consumer provides payment details during the checkout, usually a debit or credit card, which the BNPL provider verifies and authorizes for the future planned payments of the instalments.

2.7. Cash on Delivery (COD)

Cash on Delivery (COD) is a possible payment method for online shopping, and it can be selected among the others during the checkout. It allows customers to pay for goods upon delivery. Users receive the product first and pay in cash, card or other bank funds transfer options upon receipt.

3. API/SDK for Online Payments

This section describes various APIs and SDKs used by online payment platforms following a specific list of apps provided by DG-Comp, explaining how these APIs and SDKs are used for online, POS, QR, and other types of payments through different methods and functionalities.

3.1. Samsung Pay and Wallet

Samsung Pay is the Samsung app dedicated to mobile payments. It is available for selected smartphones and wearable devices such as smartwatches. Launched as a standalone service, it is now part of the Samsung Wallet app that supports storage of digital IDs, home and car keys, tickets and passes, loyalty cards, and passwords[1]. Samsung Pay supports NFC (in-store) payments, in-app and online payments. The latter can be done both on a mobile phone and a desktop computer, in this case with payment authorisation done on the smartphone.

3.1.1. Architecture

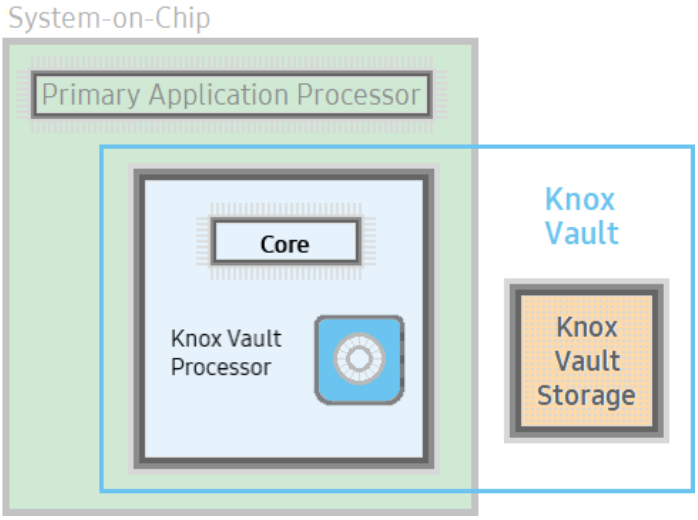
The architecture of Samsung Wallet relies on the Samsung Knox security platform, which offers a separate secure OS, fingerprint recognition to authorise operations, secure storage and rooting detection. Samsung Knox provides a Trusted Execution Environment (TEE) by leveraging on the ARM TrustZone, ARM Hypervisor Mode, and embedded Secure Elements[2]. An important component of Knox is the Vault, an isolated and tamper-proof subsystem that unlike the ARM TrustZone (which runs a separate OS on the main application processor) runs on its own processor and memory, the isolated tamper-proof Knox Vault. The picture below shows an overview of the Knox Vault component with respect to the primary application processor[3]:

[1] https://www.samsung.com/us/apps/samsung-wallet/

[2] https://docs.samsungknox.com/admin/fundamentals/whitepaper/samsung-knox-for-android/the-samsung-knox-platform/

[3] https://docs.samsungknox.com/admin/fundamentals/whitepaper/samsung-knox-for-android/core-platform-security/knox-vault/

Figure 1 Knox Vault and application processor



Source: Samsungknox.com[4])

The Knox platform offers support for the management of certificates and credentials in the devices. In particular, the Universal Certificate Management (UCM) framework supports the storage of credentials and secrets on Secure Elements, either embedded (eSE) or removable (e.g., in microSD, SIM or smart cards). Use cases based on the eSE are typically those related to payments, digital keys or identification[5].

3.1.2. API/SDK description

Samsung provides two Software Development Kits (SDKs) to integrate payment functionalities in mobile apps and web applications.

The first one is the **Samsung Pay SDK**[6], an Android SDK and application framework for integrating selected Samsung Wallet features within Android apps on Samsung devices. Those are essentially merchant apps, card issuer apps or other wallet apps offering the two following major operations:

— *In-App Payment*: to offer customers the possibility to pay for products and services using Samsung Wallet;

— *Push Provisioning*: to offer customers the possibility to add a bank card from the issuer app to Samsung Wallet.

Referring to the in-app payment use case, the merchant app is the one presenting to the user the possibility to complete a purchase by paying with Samsung Wallet. When this choice is done, the merchant app would call the APIs in the Samsung Pay SDK to initiate a transaction with Samsung Wallet app. The latter is the one providing the tokenised payment information necessary to

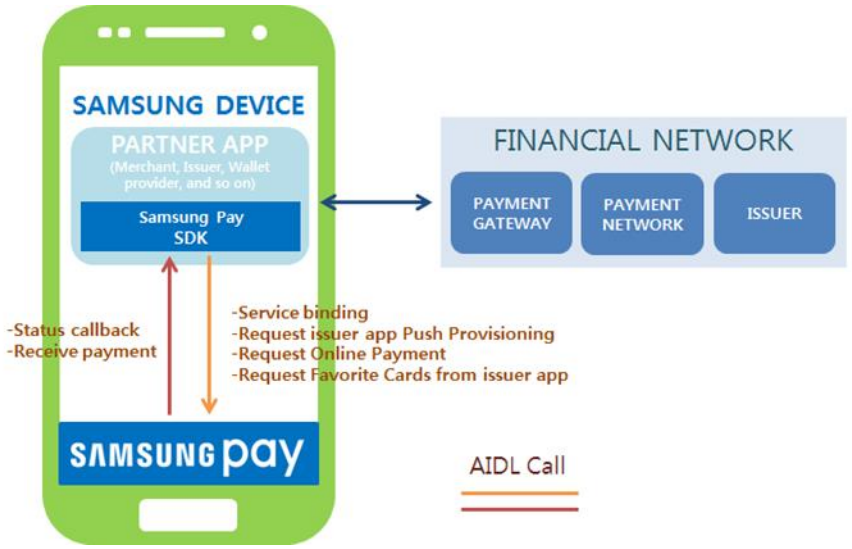
[4] https://docs.samsungknox.com/admin/fundamentals/whitepaper/samsung-knox-mobile-security/system-security/knox-vault/

[5] https://developer.samsung.com/ese/overview.html

[6] https://developer.samsung.com/pay/native/sdk-overview.html

complete the transaction and therefore performing the actual payment. The merchant app would only forward the payment information (encrypted token) to a Payment Gateway (PG), either directly (Network Token Mode) or via the Samsung-PG Interface server (Gateway Token mode). In the latter case, the token is received by the PG from the Samsung Pay app[7]. The following figure shows an overview of the Samsung Pay SDK interaction with partner apps.

Figure 2: Samsung Pay SDK interaction with partner apps



Source: Samsung Developer[8]

The second SDK is the **Samsung Pay Web Checkout**[9], a JavaScript-based web development kit to integrate the Samsung Pay payment system in a website to accept payments from Samsung Wallet. The purpose of the Web Checkout SDK is to enable users to pay for purchases on a website with the Samsung Wallet application configured on their mobile device. In other words, Web Checkout supports browser-based payments that can be done on both computers and mobile devices. In both cases the user must have a mobile device with Samsung Wallet installed. In fact, the payment process is triggered on the browser but as already mentioned at the beginning, the payment authorisation is done on the Samsung Wallet app, which ultimately processes payment information.

The SDKs mentioned above provide interaction with the Samsung Wallet that maintains control over the communication with the embedded secure element where card information is safely stored. This means that these two libraries do not enable direct communication with the eSE for the implementation of full payment process. This type of API is also not offered by the Android OS[10],

[7] https://developer.samsung.com/pay/native/inapp.html

[8] https://developer.samsung.com/pay/native/sdk-overview.html

[9] https://developer.samsung.com/pay/web/overview.html

[10] https://developer.android.com/develop/connectivity/nfc/hce#coexistence

which instead provides support for HCE[11] functionalities. HCE and SE functions can however work in parallel.

Currently Samsung provides also the **Samsung eSE SDK**[12] that gives the possibility to service providers to offer services using the embedded Secure Element. These services relate to use cases such as payments, transit and ID cards, secure storage etc. With this SDK, service providers can develop their applets to be installed in the eSE, enable communication between their client application on the device and the secure element and finally with their servers. Services are therefore managed directly by the provider on customer devices leveraging on the security features offered by the SE, and in particular its capability to perform transactions independently from the main processor.

It is worth to note here that Google has created the Android Ready SE Alliance, a collaboration between Google and Secure Element (SE) vendors[13]. Samsung is listed as one of the partners. As stated in the SE Alliance webpage "*OEMs that adopt Android Ready SE can produce devices that are highly secure and allow for remote updates to enable compelling new cases as they are introduced into the Android platform*". One of the main goals is also to "*enable a consistent, interoperable, and demonstrably secure set of applets across the Android ecosystem*".

3.1.2.1. Access and conditions

Samsung Pay SDK and Samsung Pay Web Checkout are subject to the same access rules[14]. A developer must become a Samsung Pay Partner by signing up on the Samsung Pay Portal to request access to the Samsung Pay Developers site. The access to the portal requires a Samsung business account, and in the membership request to join the developer community members should provide information about their company (e.g., contact information, type of business, size). Although not explicitly mentioned it seems therefore that individuals cannot request this membership. For the Android SDK (Samsung Pay SDK) members must also provide information about the app and set a test environment. The registration of the app requires to upload the APK file on the Samsung portal.

The Samsung eSE SDK instead requires different access modalities[15]. Apart from becoming a **Samsung Partner**, provide company information and information about the app, developer should sign a **Non-disclosure Agreement** (NDA). In this case Samsung specifies that individual developers cannot become partners for the Samsung eSE SDK.

3.2. Google Pay

Google Pay is Google's digital payment service available on both Android smartphones and wearable devices. Its core function is to enable mobile and online payments, supporting both in-store NFC payments and online payments for Android and web applications. The branding of Google's payment services has evolved over time, from "Android Pay" and "Google Wallet". Today,

[11] Host card emulation (HCE) is a technology for securing a mobile phone such that it can be used to make credit or debit transactions at a physical point-of-sale (POS) terminals.

[12] https://developer.samsung.com/ese/overview.html

[13] https://developers.google.com/android/security/android-ready-se

[14] https://developer.samsung.com/pay/native/partner-onbarding.html

[15] https://developer.samsung.com/ese/overview.html

Google Pay specifically denotes the payment processing service, while Google Wallet refers to Google's digital wallet that also supports ID storage, transit cards and event tickets[16]. In some regions, which do not include the EU, the service allows for peer-to-peer transactions.

Google Pay operates across Android and web-based systems and can be integrated through multiple development platforms.

3.2.1. Architecture

Google Pay's payment architecture employs EMV tokenization[17] for all payment transactions. When adding a new card to Google Wallet, actual payment card data is replaced with a unique token, generated by Google and the card network, providing security and privacy of payment information[18] and preventing the need to store sensitive information like an actual card number.

In-person payments rely on NFC paired with Host Card Emulation (HCE), a method that enables in-store payments without requiring direct access to a Secure Element (SE) on the device. This approach expands the range of compatible devices. A limited number of *payment tokens*, which are tokens enabling single transactions, is always pre-stored into the user's phone so that it is possible to perform a limited number of payments even in the absence of a connection.

An added level of security is provided by biometric authentication (fingerprint or face recognition) and risk-based authentication (RBA) measures for detecting and preventing fraud.

For in-app and online purchases, Google's server processes payments directly with the bank using tokenized payment information. This allows cards registered in Google Wallet to be used across any device logged into the corresponding Google account. In principle, when making payments through a desktop browser, access to the phone is not required unless it is needed for logging into the Google account (e.g., if two-factor authentication is enabled).

3.2.2. API/SDK description

Google Pay provides two specialized APIs for Android and web-based applications[19]. The purpose of these APIs is to add Google Pay as a payment option within an Android app or an existing website during the checkout phase, whenever the user is making a one-time purchase of goods or services or setting up a subscription with recurring payments.

— The *Android API* supports integration through the most common programming languages and frameworks used for Android development, including Java, Kotlin, Flutter, and React Native, enabling developers to incorporate Google Pay functionality in their own Android applications.

[16] https://www.androidpolice.com/google-pay-becomes-google-wallet-again

[17] EMV tokenization is digitizing a single physical payment card into several independent digital payments means through tokens.

[18] https://medium.com/mobilepeople/a-deep-dive-into-google-pay-and-apple-pay-d56dab7194a0

[19] https://github.com/google-pay

— The *Web API* consists of the *google-pay-button* JavaScript library[20], which helps developers add a Google Pay button to websites through a Web Component or as React and Angular components.

Financial institutions issuing cards can further integrate with the Google Pay ecosystem by requesting access to the Push Provisioning API for enabling their customers to add cards directly to Google Wallet. These institutions can also utilize the TapandPay API for enabling tap-to-pay functionality. Access to these APIs and their documentation is restricted to such entities after Google's approval.

3.2.3. API/SDK access and conditions

Developers can integrate Google Pay into an Android app or website using a test environment with any developer account21,22. To switch to production, a merchant profile must be created, and the implementation must be reviewed and approved by Google.

Applications and websites using Google Pay must adhere to brand guidelines[23,24] for correctly displaying Google's brands and payment buttons. Following these guidelines is mandatory to obtain approval for production access. In addition, there are a few optional best practice and suggestions, such as making Google Pay the primary or sole payment option, which do not affect approval if not implemented.

As previously mentioned, access to Push Provisioning and TapandPay APIs is restricted to financial institutions that have requested and obtained approval from Google.

3.2.4. API/SDK comparison

In terms of use cases and features, Google Pay is broadly comparable to Samsung Pay: both support in-person, in-app, and online payment options. However, while Google Pay targets the entire Android ecosystem, Samsung Pay is specifically designed for Samsung devices. The Google Pay SDK documentation is intended for implementing Google's payment services within Android applications and as a checkout option on websites. It is unclear if any parts of the Google Pay SDK are reserved for Google's internal use and restricted from third-party developers, although there are no apparent indications suggesting such limitations.

3.3. Amazon Pay

Amazon Pay is a platform-independent online payment service. It enables its partners (merchants) to provide a payment mechanism through the Amazon ecosystem on their respective websites. The merchants' customers rely on the information they have stored in their Amazon account. In SEPA

[20] https://github.com/google-pay/google-pay-button

[21] https://developers.google.com/pay/api/web/guides/test-and-deploy/integration-checklist

[22] https://developers.google.com/pay/api/android/guides/test-and-deploy/integration-checklist

[23] https://developers.google.com/pay/api/android/guides/brand-guidelines

https://developers.google.com/pay/api/web/guides/brand-guidelines

customers can complete their transactions with credit cards (Visa, MasterCard, American Express), debit cards (Visa Electron, Delta, Maestro), or direct debit with bank accounts from SEPA countries[25]. Therefore, there is no need for the partners to handle any authentication, storage of payment information, or payment processing. The integration process can be done through Amazon's commerce partners, such as BigCommerce, Adobe Commerce, or Woo[26], through payment service providers such as Optile, Computop, or Payone[27], or through five, Amazon-provided, open-source SDKs[28] or their API[29].

[25] https://pay.amazon.eu/help/201754650

[26] https://pay.amazon.eu/integration#commerce-solution-provider

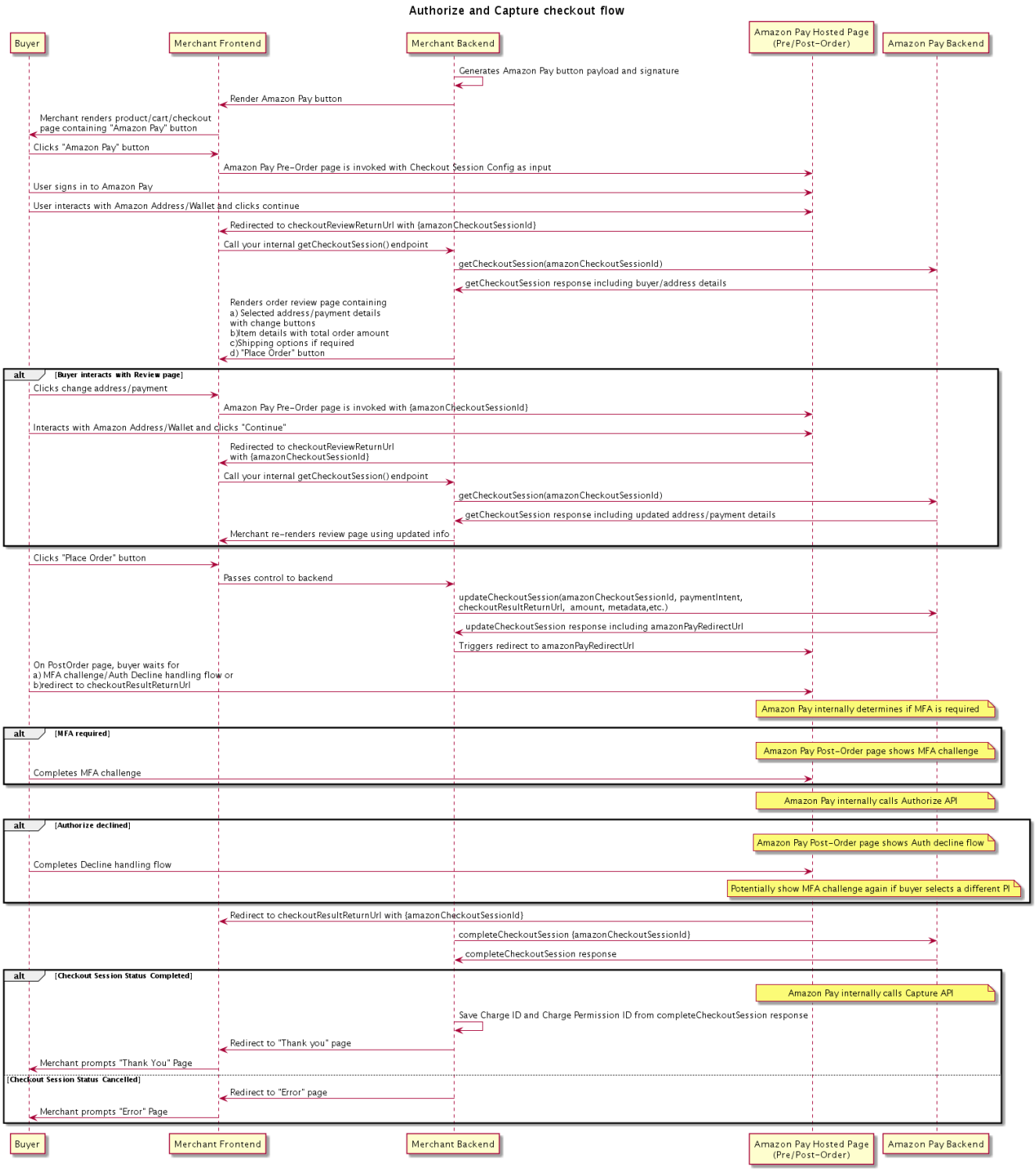
[27] https://pay.amazon.eu/integration#payment-service-provider

[28] https://github.com/orgs/amzn/repositories

[29] https://developer.amazon.com/docs/amazon-pay-api-v2/introduction.html

3.3.1. Architecture

Figure 3: Sequence diagram of how each payment is processed in Amazon Pay



Source: Amazon Developer Portal[30]

In their documentation, Amazon provides a diagram (image above)[31][32] on how each payment is processed. A summary of the process:

1. The buyer clicks on the Amazon Pay button. (merchant)
2. The buyer signs in to their Amazon account. (Amazon)
3. The buyers selects/modifies their address/payment information. (Amazon)
4. The buyer clicks on the place order button. (Amazon)
5. MFA challenge or card declined. (Amazon)
6. Thank you or error page. (merchant)

3.3.2. API/SDK Description

Amazon Pay provides two integration models – a high-level integration through Amazon's commerce and payment service partners, or a lower-level integration through Amazon's API/SDKs. Merchants can register a developer account to access the Sandbox environment which can be used for testing purposes. For production access, merchants must open a merchant account[33].

3.3.2.1. Commerce and Payment Service Partners

Integration with commerce partners[34] is partner-dependent; however, they all share a similar approach. The merchant installs a plug-in or a library component of the selected commerce partner and then connects it to Amazon Pay via an API key that is either transferred manually by the merchant, or automatically, depending on the selected commerce partner. Integration with a payment service partner[35] is more customised to the merchant and includes integration of a suite of different payment solutions, where Amazon Pay is included.

3.3.2.2. API/SDKs

Amazon provides five different SDKs for PHP[36], Ruby[37], Java[38], C#[39], and JavaScript[40]. All are open source, licensed under Apache-2.0 licence, and published on Amazon's GitHub[41] code repository platform. Each of the SDKs provides an implementation of all API calls required to fully integrate Amazon Pay within the target application and as such serve as a complementary utility to the also provided low-level API[42].

---

[30] https://developer.amazon.com/docs/amazon-pay-checkout/overview.html
[31] https://developer.amazon.com/docs/amazon-pay-checkout/overview.html
[32] https://developer.amazon.com/docs/amazon-pay-paymentmethodonfile/overview.html
[33] https://pay.amazon.eu/signup
[34] https://pay.amazon.eu/integration#commerce-solution-provider
[35] https://pay.amazon.eu/integration#payment-service-provider
[36] https://github.com/amzn/amazon-pay-api-sdk-php
[37] https://github.com/amzn/amazon-pay-api-sdk-ruby
[38] https://github.com/amzn/amazon-pay-api-sdk-java
[39] https://github.com/amzn/amazon-pay-api-sdk-dotnet/
[40] https://github.com/amzn/amazon-pay-api-sdk-nodejs
[41] https://github.com/orgs/amzn/repositories?language=&q=pay&sort=&type=all
[42] https://developer.amazon.com/docs/amazon-pay-api-v2/introduction.html

3.3.2.3. Access and Conditions

Merchants' access to the API for any of the above listed methods is dependent upon a registration of an Amazon Payments merchant account[43]. Within the EU Amazon Pay supports businesses located in only selected countries[44] and includes a special condition for Ireland of having a yearly turnover of more than 3 million euros[45]. The merchants are also subject to Acceptable Use Policy[46] which prohibits certain items and activities from being purchased with Amazon Pay.

3.4. Blik

Blik is an account-to-account mobile payment solution that operates in Poland and is integrated into the mobile applications of nineteen supported banks. All participating banks have a mobile application available for iOS and Android, seven banks also have applications available for HarmonyOS.

Blik enables their customers to make online payments, pay in physical stores, send money via a phone number, and withdraw and deposit money at ATMs. Online payments are authorised by a *Blik code* (a numerical 6-digit token valid for 120 seconds) a customer provides to the online store. If implemented by the merchant, online payments can also be processed without the *Blik code*. To enable such payments for a given online store the buyer must first complete a purchase with the *Blik code*. At the time of payment confirmation, there is an additional option for the buyer to *save* this online store. If they choose to do so, the *Blik code* is not required with further purchases from the same online store. Physical store payments are carried out with a *Blik code* or contactless payment (HCE) that requires a screen lock (PIN or biometrics). Sending money via phone number is available only to other users of Blik and is done by selecting a contact from the sender's contacts list or by inserting their phone number. Withdrawing and depositing money at ATMs is again executed by providing the *Blik code* to the ATM.

3.4.1. Architecture

Blik defines three different payment processes: payments with *Blik code,* payments without *Blik code,* and recurring payments.

[43] https://pay.amazon.eu/help/202153180

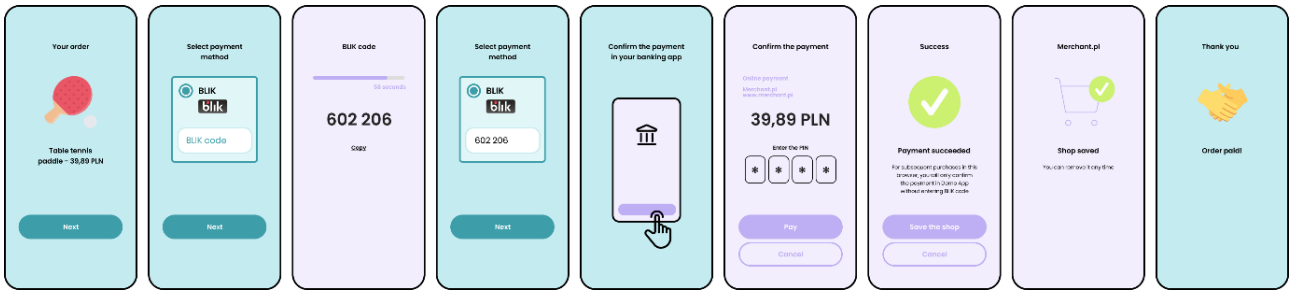
[44] https://pay.amazon.eu/help/201810860#toc_In-which-countries-can-my-business-be-located-

[45] https://pay.amazon.eu/signup

[46] https://pay.amazon.eu/help/6023

3.4.1.1. Payment With Blik Code[47]

Figure 4: Payment with *Blik code* flowchart



Source: blik.com[48]

Summary of the above flowchart:

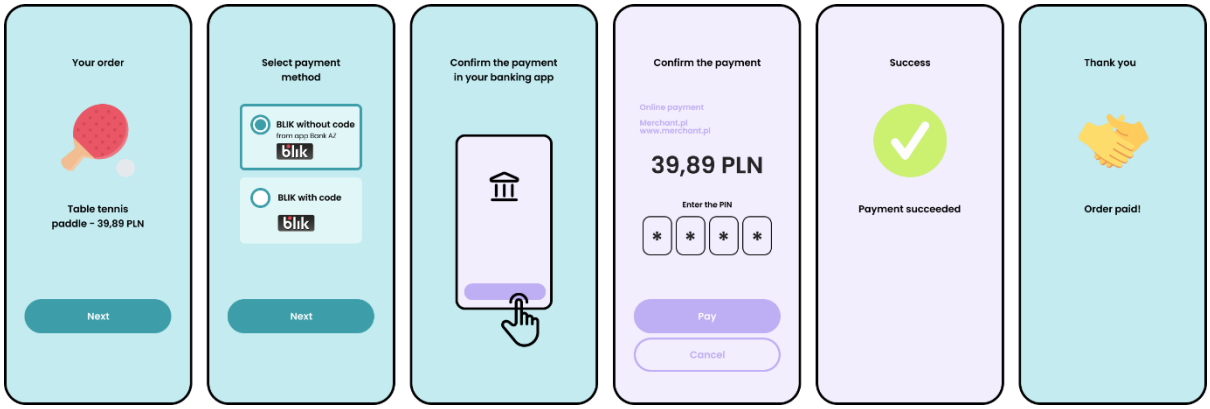
1. The buyer selects Blik as their payment method. (merchant)
2. The buyer takes note of the *Blik code*. (banking app)
3. The buyer inserts the *Blik code* in the payment method screen. (merchant)
4. The buyer confirms the payment in the banking app. (banking app)
5. The buyer optionally saves the merchant store to use without the *Blik code*. (banking app)
6. Thank you page. (merchant)

[47] https://blik.com/lp/payment-without-code/Level-0-+-OneClick.141951221.html

[48] https://www.blik.com/lp/payment-without-code/Level-0-+-OneClick.141951221.html

3.4.1.2. Payment Without Blik Code[49]

Figure 5: Payment without *Blik code* flowchart



Source: Blik.com[50]

Summary of the above flow chart:

1. The buyer selects *Blik without code* as their payment method. (merchant)
2. The buyer confirms their payment in their banking application. (banking app)
3. Thank you page. (merchant)

For selected banks, the second step can also be omitted.

3.4.1.3. Recurring Payments[51]

Blik also includes a capability to handle recurring payments such as subscription service fees. There are three different implementation models each designed for a specific use case. In Model A, transactions are processed at a fixed time interval for a fixed amount and do not require customer's confirmation. Recurring payments under this model also have a fixed expiration date after which they have to be renewed. This model is purposed for flat-fee subscription businesses, such as streaming services. Next, Model M, varies in the time frequency and monetary amount of the transactions, however, each transaction must be confirmed by the customer. Recurrence under this model can have a set expiration date or set as indefinite or until cancellation. Model M is designed for payment of household bills and usage-based services, e.g. cloud computing. Lastly, Model O, shares all properties of Model M, apart from a lack of requirement for confirmation by the customer. Its usage is anticipated for services provided by mobility apps, or content providers with functionality that allows additional content for purchase.

[49] https://blik.com/lp/payment-without-code/Payments-without-the-BLIK-code.141951211.html

[50] https://www.blik.com/lp/payment-without-code/Level-0-+-OneClick.141951221.html

[51] https://blik.com/lp/reccuring-payments/Introduction.141951041.html

3.4.2. API/SDK Access and Conditions

Blik itself is integrated in mobile applications of several Polish banks. These integrations details are not publicly available. Therefore, Blik does not provide any API/SDK to enable merchants to integrate Blik's solution with their business. Instead, merchants must employ a service of Blik's integration partners[52] for integration of both, physical and online payments. Implemented BLIK functionalities are integrator dependent. Complete list of integrators currently includes 37 providers[53]. The amount of information on integration guidelines and terms of service available online varies between integrators. Most only provide a contact form for more information on Blik integration (Autopay[54], Dotpay[55], Eservice[56], Paysafe[57], Neopay[58]), others provide pricing information (Adyen[59], Paynow[60]) while some state specific conditions, e.g. countries where Blik can be accepted — EU and other European countries (Stripe[61]).

There are two possible implementation methods of BLIK payments — one that includes redirection to the eblik service and another that enables BLIK payments without leaving the online store (Level 0 + OneClick). First approach does not require a certification process as the payment is carried out by the eblik service. There is no further information available online on how the eblik service operates. The second approach requires a certification process carried out in two steps[62] — first for the integration partners (backend certification — Grade 1) and second for the merchants (frontend certification — Grade 2). Grade 1 certification information is not publicly available as it is only shared with the integrators. Grade 2 certification is available online in a form of a checklist of required and recommended UI elements for the merchants to implement and differs between payments without code[63] and recurring payments[64]. Payments with *Blik code* do not have a dedicated document explaining its functionalities and certification process but are mentioned in the documentation about payments without code. Required UI elements include specifications about naming schemes, *Blik code* input fields look and feel, and success and error pages, while recommendations include further information on various error messages and other cosmetic guidelines. There is no information available online on how and by whom the certification process is carried out.

---

[52] https://blik.com/en/partners#id-acquirers

[53] https://blik.com/lp/payment-without-code/List_of_Agents_Supporting_a_Given_Integration_Model.html

[54] https://autopay.eu/products/payments/blik

[55] https://www.przelewy24.pl/en/payment-methods/blik

[56] https://www.eservice.pl/en/mobile-payments-blik

[57] https://www.paysafe.com/en/local-payment-methods/

[58] https://neopay.online/lv/services/blik#

[59] https://www.adyen.com/payment-methods/blik#features

[60] https://www.paynow.pl/

[61] https://docs.stripe.com/payments/blik#request-blik-capabilities-for-your-connected-accounts

[62] https://blik.com/lp/payment-without-code/Certification-process.141951236.html

[63] https://blik.com/lp/payment-without-code/Level-0-Oneclick-Checklist.141951240.html

[64] https://blik.com/lp/reccuring-payments/Level-0---UX-Certification-Checklist.141951142.html

3.5. PayPal

PayPal is an online payment platform to send and receive money securely, using a mobile app or website. PayPal acts as an intermediary between a user and a bank. The user adds a bank account, a debit and/or a credit card to their PayPal account which can be used when making an online payment or for topping up the PayPal account. The transactions are processed by PayPal instead of the bank, and when the user receives money, they can either stay in the PayPal account or transferred to the bank account. It has over 400 million user and merchant accounts globally[65]. The PayPal mobile app is available for Android and iOS and does not have any special hardware requirements as it processes transactions online similarly to the website version.

3.5.1. Architecture

PayPal processes payments according to the following transaction flow[66]:

1. The merchant creates a PayPal merchant account to accept payments from customers
2. The merchant links this PayPal merchant account to their website
3. The customer selects the item(s) they wish to buy on a website, reviews the order and click the Pay Now button
4. The customer is transferred to the payment page where they can log in to pay with a PayPal account or choose to pay with a credit/debit card
5. If the customer selects the card option, they enter their card details and PayPal sends a request to the issuing bank to check if there is enough credit
6. The bank approves or rejects the request
7. PayPal responds to the customer whether the payment was accepted or not
8. If the payment was accepted, the issuing bank sends the funds to the merchant PayPal account; the merchant can then select to leave it in the PayPal account or transfer it somewhere else (e.g., the merchant's bank account).

[65] https://about.pypl.com/who-we-are/history-and-facts/default.aspx

[66] https://www.paypal.com/us/brc/article/how-paypal-works-for-sellers

3.5.2. API/SDK Description

3.5.2.1. PayPal REST APIs[67]

PayPal APIs are based on REST and authentication is performed with OAuth 2.0 access tokens. They are used to create and manage payment functions of websites. These REST APIs are grouped into collections, according to the functionalities that they provide. The following table lists the current collections and their functionalities:

Table 1: List of the current collections of PayPal REST APIs and their functionalities.

Collection	**Functionalities**
Add Tracking	Manage tracking information on your PayPal transactions
Catalog Products	Create products, which are goods and services, to then use as product offerings in subscriptions.
Disputes	Manage customer initiated disputes, chargebacks, or bank reversals.
Identity	Get user profile information and manage user account settings.
Invoicing	Create, send, and manage invoices, including tracking invoice payments.
Orders	Create, update, retrieve, authorize, and capture orders.
Partner Referrals	Add PayPal seller accounts to your platform to connect your seller with PayPal before they accept a buyer payment.
Payment Experience	Create web experience profiles to customize payment flow experiences from cart to buyer payment.
Payments	Use in conjunction with the Orders API to authorize payments, capture authorized payments, refund payments that have already been captured, and show payment information.
Payouts	Make payments or send commissions, rebates, rewards, and general disbursements to multiple PayPal or Venmo recipients.
Referenced Payouts	Disburse held funds captured in a delayed payment from a buyer to your seller.
Subscriptions	Create subscriptions that process recurring PayPal payments for physical or digital goods, or services.
Transaction Search	Get the transaction history for a PayPal account.
Webhooks	Subscribe to and manage your application's webhook events.

Source: PayPal[68]

[67] https://developer.paypal.com/api/rest/

[68] https://developer.paypal.com/api/rest/current-resources/

3.5.2.2. JavaScript SDK[69]

The JavaScript SDK provided by PayPal provides software building tools to enable developers to create uniform checkout interfaces. This SDK can be used to render the following objects:

— **buttons**: payment buttons that automatically show all the buttons that are eligible in the respective page. This component has several options, such as style for button customisation (including colour, shape and size), message for customising the text appearing on the button and createOrder for setting up the transaction details. According to the PayPal SDK documentation: "*The buttons that show up are decided automatically, based on a range of factors, including: (i) Buyer country, (ii) Device type, (iii) Funding sources the buyer has opted to see. As a result, each buyer sees a unique combination of buttons. Pay Later offers differ by country and have different buttons*". Regarding buyer country[70]: "*The buyer country determines which funding sources are eligible for a given buyer. Defaults to the buyer's IP geolocation. Any country that you can pass as a locale is a valid buyer country*". However, it is noted that this option is used only in the sandbox for testing and not in production. The documentation includes also a table listing the available payment methods per country[71], as well as card brands and currencies supported in each country[72]. It appears that the device type[73] (mobile, desktop, tablet etc.) only affects the design, position[74] and layout[75] of the button and has nothing to do with payment methods. The funding sources that the buyer has opted to see (enable-funding parameter) is an optional parameter that bypasses the desired payment methods that are set based on the buyer's country[76]. The way available buttons are displayed to the end user is influenced by the API. More specifically, according to the source code publicly available[77], the API promotes PayPal instead of other payment options. In Section paypal.Marks().render(container) in the link above, there are 2 payment methods for users: PayPal and alternate (any other payment option). These two options are represented as radio buttons where the user can select one or the other (in the following note the difference between "radio button" and "button"). When the page loads, the PayPal radio button is preselected ("checked" in line 3) and the alternate radio button not (line 8). When the PayPal radio button is selected then only the PayPal (payment) button is displayed in the page (lines 28-33). If the user selects the alternate radio button, then the alternate button is displayed (lines 37-41). When the page loads, the alternate method button is hidden by default (line 47-48). In practice, when the page loads the user will only see the PayPal payment button by default and has the option to click on the alternate radio button to display the alternate payment method button.

[69] https://developer.paypal.com/sdk/js/

[70] https://developer.paypal.com/sdk/js/configuration/#disable-funding

[71] https://developer.paypal.com/docs/multiparty/payment-methods/

[72] https://developer.paypal.com/docs/checkout/advanced/eligibility/#eligible-cards-and-currencies

[73] https://developer.paypal.com/docs/api/payments/v2/

[74] https://developer.paypal.com/docs/log-in-with-paypal/customize/button-design-guide/

[75] https://developer.paypal.com/sdk/js/reference/

[76] https://developer.paypal.com/beta/apm-beta/additional-information/js-sdk-params-payment-fields/

[77] https://developer.paypal.com/sdk/js/reference/#link-marks,

— **messages**: this component is used to show Pay Later messages on the website. Pay Later is a PayPal feature that allows customers to make purchases over instalments without interest.

— **marks**: this component is used to automatically show images for PayPal payment options (including PayPal, Venmo, Pay Later, debit and credit cards).

— **card-fields**: they are used by third parties to accept and save credit and debit card details without handling card information; PayPal hosts card-fields and handles all security and compliance aspects of processing cards.

— **funding eligibility**: according to the PayPal JavaScript SDK eligibility logic, the payment buttons automatically render all eligible buttons in a single location on a webpage so that only the appropriate buttons for the current buyer show up. Also, it seems no possible for the merchant to add third-party payment options not already supported and listed in the SDK to the webpage.

3.5.2.3. Mobile SDK[78,79]

The Mobile SDK is used to accept PayPal, credit, and debit card payments. There are three types of payments using this SDK:

— Card payments: card-fields can be used for paying with cards

— Native payments: a PayPal checkout page is launched directly within the app instead of a popup

— Web payments: a checkout page is launched in a browser within the app.

The mobile SDK comprises two SDKs that both have similar functionality: the Android SDK and the iOS SDK.

3.5.2.4. Other APIs and SDKs

There are also some other APIs and SDKs provided by PayPal that are use case specific. These include:

— Zettle[80]: it provides APIs and SDKs for integrating payment methods and point of sales (POS) services. A basic integration flow using Zettle is as follows: (a) a developer creates an app using Zettle APIs/SDKs, (b) a merchant uses an existing or creates a new account at zettle.com, and (c) the merchant authorizes the developed app which is hosted on the merchant's site. Note that Zettle is supported in some of the EU countries.

— Payouts API[81]: this API enables the developer to send payouts programmatically to recipients. The typical flow is as follows, the API: (a) validates the request and processes the payout, (b) sends a status report back to the developer or the app, and (c) notifies the recipients that they have a payment.

---

[78] https://developer.paypal.com/docs/checkout/advanced/android/

[79] https://developer.paypal.com/docs/checkout/advanced/ios/

[80] https://developer.zettle.com/docs/api

[81] https://developer.paypal.com/docs/payouts/standard/integrate-api/

— Donate SDK[82]: with this SDK, users click on a PayPal Donate button which shows a pop-up window that can be used to make donations. The difference from a normal PayPal transaction is that with the Donate button the user is kept on the website instead of redirected to a donation page on PayPal.

3.5.3. API/SDK access and conditions

PayPal has the following policy on using APIs:

— No account is needed to try REST APIs in Postman;

— A free PayPal developer account is needed to test US integrations;

— A PayPal Business account is needed to use integrations in a production environment or test integrations outside the US.

For developers, the PayPal Developer Agreement[83], the PayPal Developer Testing Agreement[84], and the PayPal Sandbox User Agreement[85] apply.

To gather information about the actual details required for accessing the APIs, the official sign-up process to create a new PayPal account was followed and the findings are summarised below. Overall, one developer and three business accounts were created. A free PayPal developer account can be created by providing some personal information, such as email address, mobile phone number and name. To create a PayPal Business account some information about the account holder and the business is needed depending on the type of the business, as follows:

— Single-owner company: owner name, date of birth, home address, social security number (SSN) and tax ID number (TIN). Also, information about the company is requested: business name, currency, type of product or service offered, monthly sales volume and company address. To confirm the account holder identity, it is requested to upload a photo ID document. Note that the online form accepts only US-based addresses even if it is allowed to select a different country as incorporation place.

— Company (corporation, multi-member limited liability company, or government): business legal name, currency, establishment and incorporation dates, type of product or service offered, monthly sales volume. To prove the accuracy of some of the provided details (such as company name and fiscal number) it is requested to upload official documents, such as bank statements or tax return forms. In addition, personal information about the account holder is requested, such as name, date of birth, home address and part of the SSN. Note that the online form accepts only US-based addresses even if it is allowed to select a different country as incorporation place.

[82] https://developer.paypal.com/sdk/donate/

[83] https://www.paypal.com/us/legalhub/paypal/xdeveloper-full?locale.x=en_US

[84] https://www.paypal.com/us/legalhub/paypal/devtest-agreement?locale.x=en_US

[85] https://www.paypal.com/us/legalhub/paypal/xsand-full?locale.x=en_US

— Non-profit: organisation name, Doing Business As (DBA) name of the organisation, currency, dates of establishment and incorporation, average monthly donations, phone number, employer identification number (EIN) and organisation address. In addition, personal information about the account holder is requested, such as name, date of birth, home address and part of the SSN. To prove the accuracy of business name and TIN or EIN numbers it is requested to upload official documents, such as bank statements or tax return forms. Note that the online form accepts only US-based addresses even if it is allowed to select a different country as incorporation place.

3.6. Vipps MobilePay

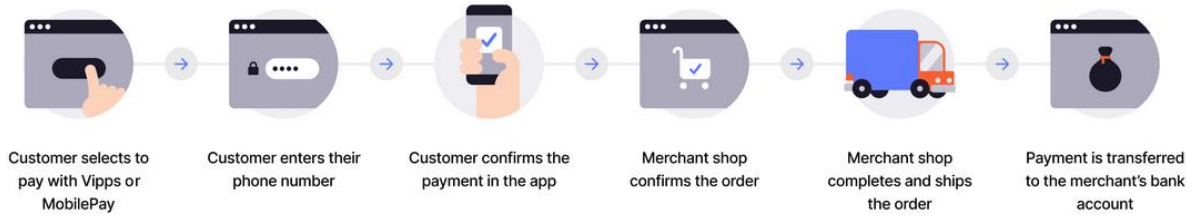
Vipps MobilePay is a popular mobile payment platform operating in Nordic countries. It is the result of a merge between Vipps, founded in Norway, and MobilePay, founded in Denmark. Nowadays it operates in Denmark, Norway, Finland and with some banks in Sweden. MobilePay was first introduced in 2013 by Danske Bank and it allows payments using a mobile app. It allows person-to-person as well as online payments and has more than 12 million users[86]; also, more than 400 thousand shops and stores are part of the platform. In terms of transactions, more than 2 million are executed daily and more than 1 billion annually. There are user apps both for Android and iOS, as well as mobile point-of-sale (POS) apps for merchants. The only requirement for a mobile POS is a smart device and the app. It is possible for the user to pay using Near Field Communication (NFC) or QR codes.

3.6.1. Architecture

The main payment types when using Vipps MobilePay are: (a) online payments and (b) in-store payments.

In online payments, the following flow is recommended:

Figure 6: Payment sequence using Vipps MobilePay: online payments



Source: Vipps MobilePay[87]

For in-store payments, the flow is as follows:

[86] https://vippsmobilepay.com/en/about#numbers

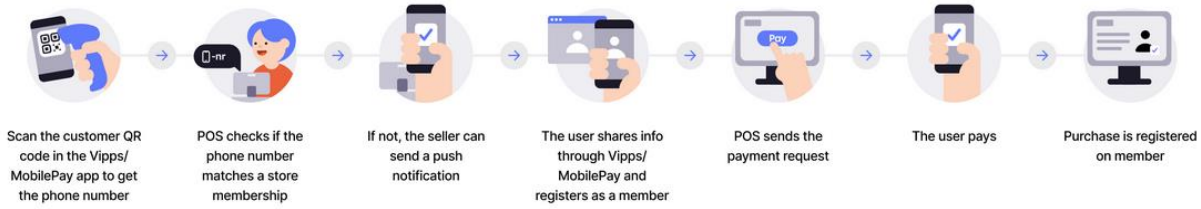
[87] https://developer.vippsmobilepay.com/docs/recommended-flows/online/

Figure 7: Payment sequence using Vipps MobilePay: in-store payments

source: Vipps MobilePay[88]

There are also some slightly modified versions of the above flow for different situations. For example, for in-store payments with loyalty cards, the flow becomes:

Figure 8: Payment sequence using Vipps MobilePay: in-store payments with loyalty cards



source: Vipps MobilePay[89]

Other more specific use cases include electric vehicle charging:

Figure 9: Payment sequence using Vipps MobilePay: electric vehicle charging

Source: Vipps MobilePay[90]

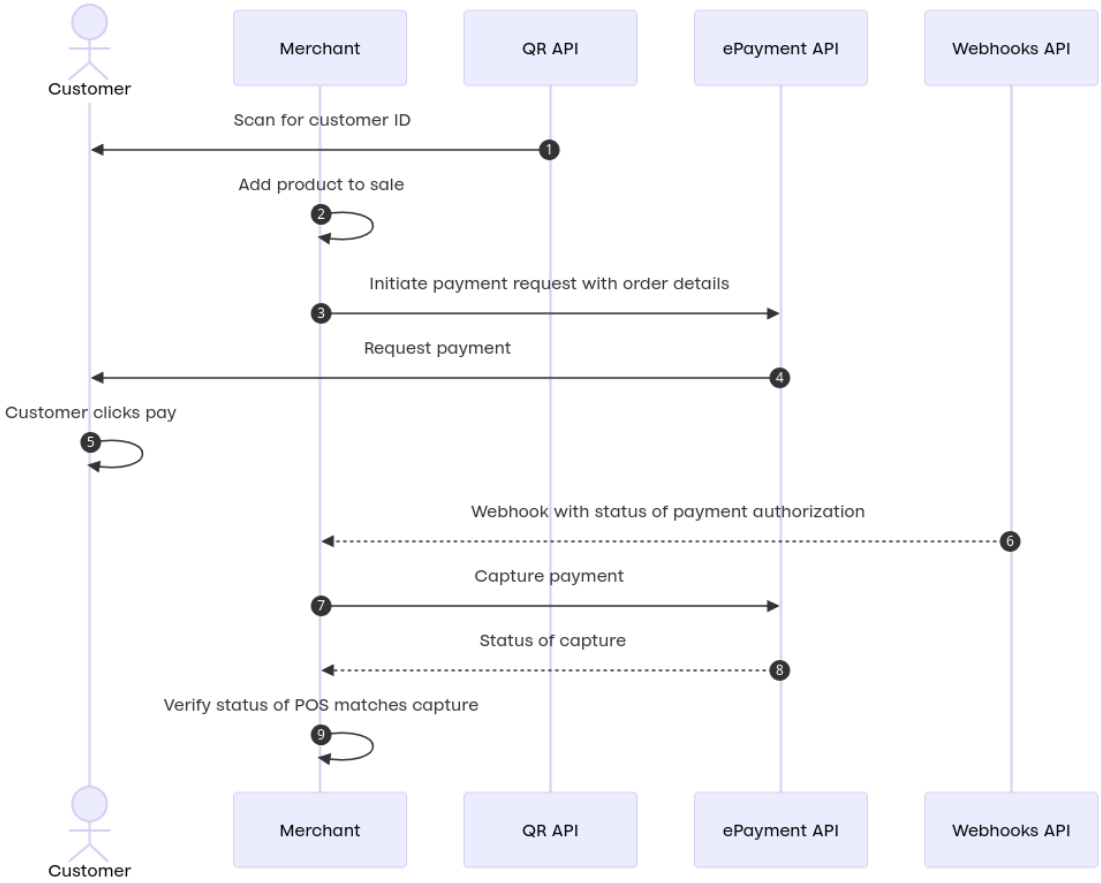
[88] https://developer.vippsmobilepay.com/docs/recommended-flows/in-store/

[89] https://developer.vippsmobilepay.com/docs/recommended-flows/loyalty-in-pos/

[90] https://developer.vippsmobilepay.com/docs/recommended-flows/ev-charging/

Or paying a taxi; first the customer scans a QR code placed in the taxi and then the following flow is executed:

Figure 10: Payment sequence using Vipps MobilePay: taxi payment through QR codes

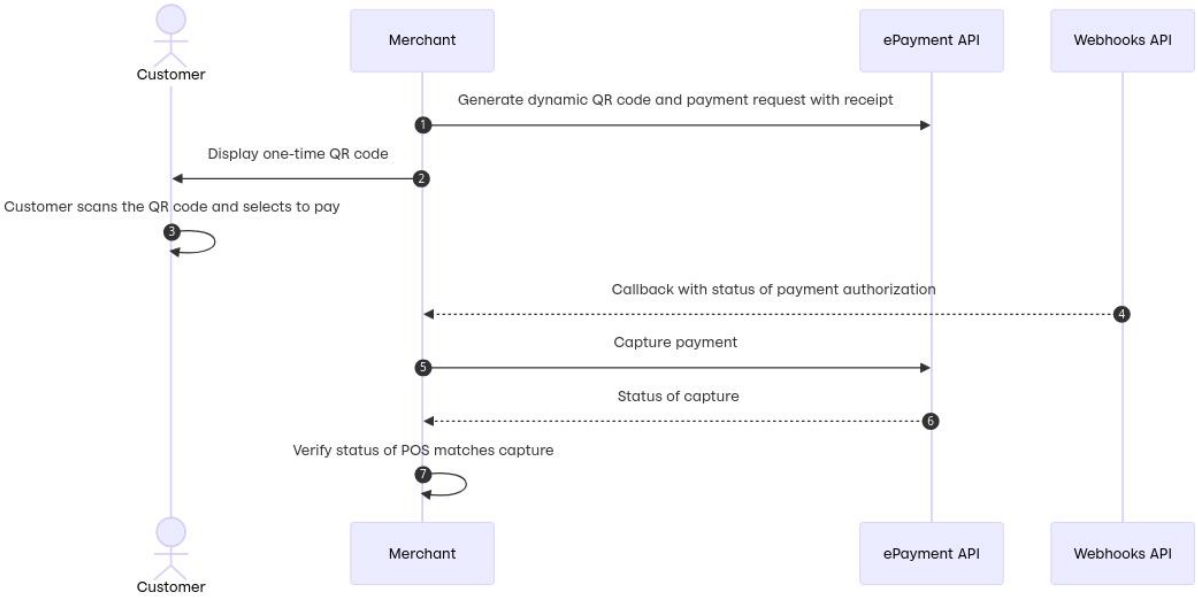


Source: Vipps MobilePay[91]

[91] https://developer.vippsmobilepay.com/docs/recommended-flows/taxi-companies/scan-qr-to-payment-page/

The following flow is a payment which can be used in vending machines. In this case, a one-time QR is produced:

Figure 11: Payment sequence using Vipps MobilePay: vending machines payment through QR codes

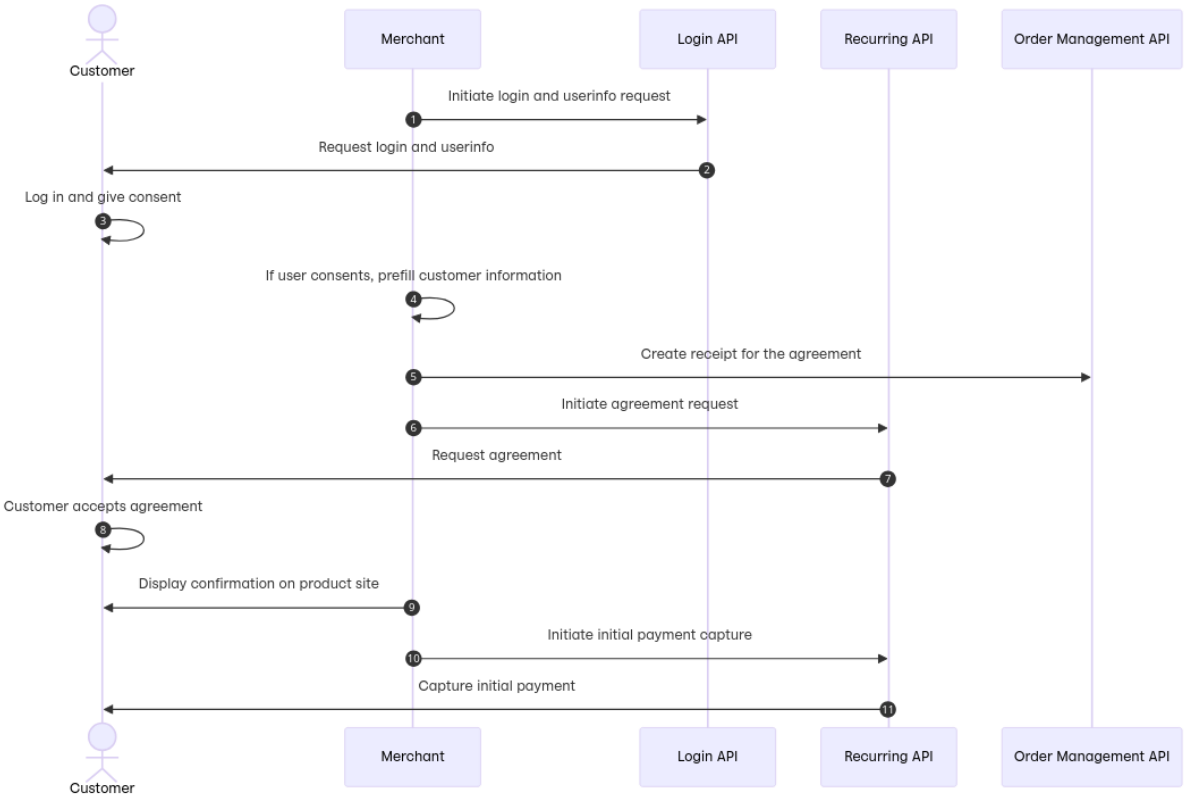


Source: Vipps MobilePay[92]

[92] https://developer.vippsmobilepay.com/docs/recommended-flows/vending-machines/one-time-payment/#sequence-diagram

Another typical scenario is for the customer to buy a subscription with recurring payment; the sequence of actions for signing up to the subscription and the initial charge is as follows:

Figure 12: Payment sequence using Vipps MobilePay: subscription purchase with recurring payments – initial charge

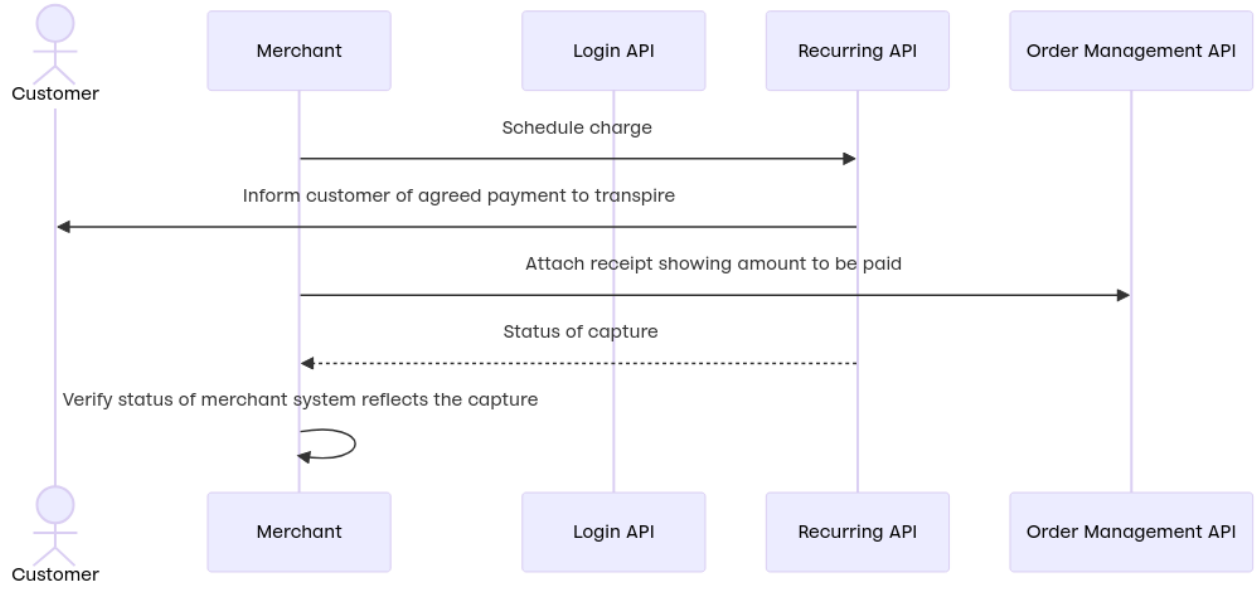


Source: Vipps MobilePay[93]

[93] https://developer.vippsmobilepay.com/docs/recommended-flows/recurring-and-login/#sequence-diagrams

Whereas yearly charges thereafter:

Figure 13: Payment sequence using Vipps MobilePay: subscription purchase with recurring payments – initial charge – yearly charges

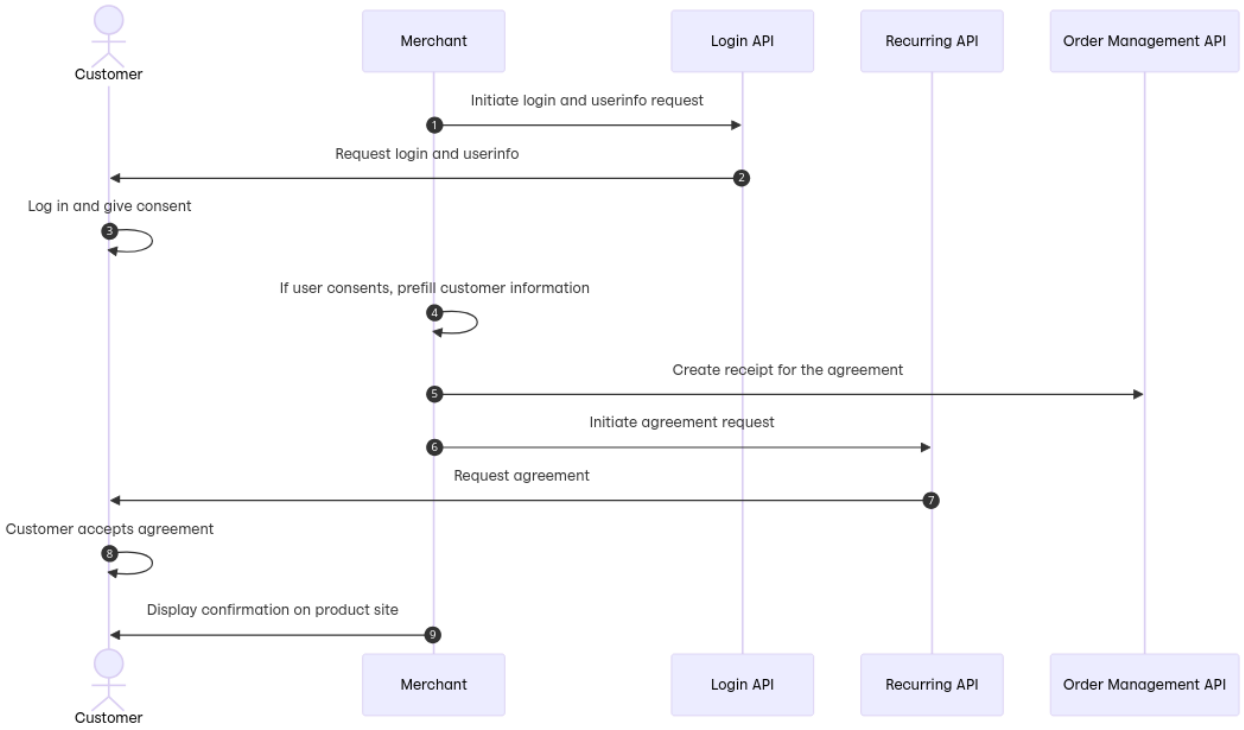


Source: Vipps MobilePay[94]

[94] https://developer.vippsmobilepay.com/docs/recommended-flows/recurring-and-login/#sequence-diagrams

Another case of recurring payments is parking. Here there are two cases, the pay-as-you-go plan:

Figure 14: Payment sequence using Vipps MobilePay: parking payment with a pay-as-you-go plan

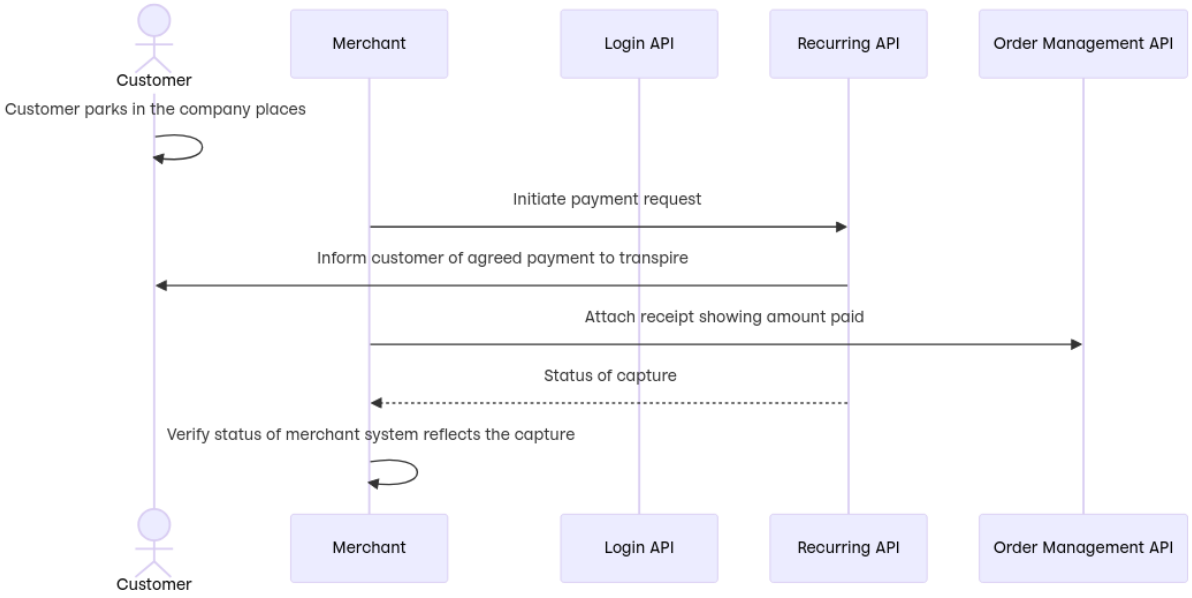


Source: Vipps MobilePay[95]

[95] https://developer.vippsmobilepay.com/docs/recommended-flows/parking/#sequence-diagrams

And the daily charging plan:

Figure 15: Payment sequence using Vipps MobilePay: parking payment with a daily charging plan



Source: Vipps MobilePay[96]

3.6.2. API/SDK Description

The following table lists all Vipps MobilePay APIs and their functionalities:

Table 2: List of all Vipps MobilePay APIs and their functionalities

API	**Functionalities**
Checkout	Through the Checkout API you can sell more with an all-in-one online checkout. Create an online checkout session where you can handle payment processing, shipping, and gathering user information.
Payment integration	Add Vipps MobilePay to your online payment setup. Perform online, POS, QR, and other types of payments using Vipps, MobilePay, or cards.
Login	Sign-up and login users online.
Recurring payments	Let returning customers pay automatically.
Access token API	Obtain an access token to authorize all other API requests. (Included with all the main products listed above.)
Check-in API	Communicate with your customer through the mobile app at the Point of Sale. (Included with all main products.)

[96] https://developer.vippsmobilepay.com/docs/recommended-flows/parking/#sequence-diagrams

Management API	Manage sales units, get information about merchants, or place orders on behalf of merchants.
Order Management API	Enrich your transactions by attaching images and information. (Included with Payment Integration and Recurring.)
PSP API	(For partners) Initiate and manage the status of PSP payments.
QR API	Generate and administer QR codes that redirect the user to a URL. (Included with Payment Integration and Recurring.)
Report API	Retrieve data about orders and settlements. (Included with Payment Integration and Recurring.)
Userinfo API	Get user consent for access to their personal information (e.g., phone number). (Included with Payment Integration, Recurring, and Login.)
Webhooks API	Register for event notifications through webhooks. (Included with Payment Integration.)

Source: Vipps MobilePay[97]

The following table lists the SDK functionalities supported by Vipps MobilePay:

Table 3: List of the SDK functionalities supported by Vipps MobilePay

Type	**Functionality**
Configuration and authorization	Authentication, getting authorization
Checkout	Create checkout session and get session info
ePayment	Create and manage payments, handle payment states and modifications
Recurring	Create and manage Recurring payments
Webhooks	Work with webhooks

Source: Vipps MobilePay[98]

3.6.3. API/SDK access and conditions

The use of Vipps MobilePay APIs is based on API keys; there are two separate sets of API keys: (i) API keys for the test environment and (ii) API keys for the production environment. All merchants with API access have access to the test environment. API keys are assigned to a sales unit, and each sales unit must have the same country code as the merchant. Thus, each sales unit has a unique set of API keys, and if a merchant has many sales units, they will use different keys for each one. Furthermore, there are different types of API keys as follows:

[97] https://developer.vippsmobilepay.com/docs/APIs/

[98] https://developer.vippsmobilepay.com/docs/SDKs/

— Merchant keys. They provide access to the common APIs (that is, the first table above) to which the merchant has access. These are used by smaller partners[99] who don't have partner keys.

— Management keys. They provide access to the Management API only. They are useful when the partner does not have partner keys, or for reasons of security, they cannot use them.

— Partner keys. They provide access to most of the common APIs from the first table above, allowing partners to act on behalf of all their merchants. Note that only partners of Partner Plus type and above qualify for such keys (roughly with annual revenue of more than 1M euro and more than 30 merchants).

— Accounting keys. They provide access to the Report API only. These cannot be used to act on behalf of merchants, or to access the Management API.

The different types of API keys provide access to the APIs as follows:

Figure 16: Mapping between different types of API keys and access to the APIs

API keys	Management API	Main APIs	Report API
Merchant keys	✅	✅	✅
Management keys	✅	❌	❌
Partner keys	✅	✅	❌
Accounting keys	❌	❌	✅

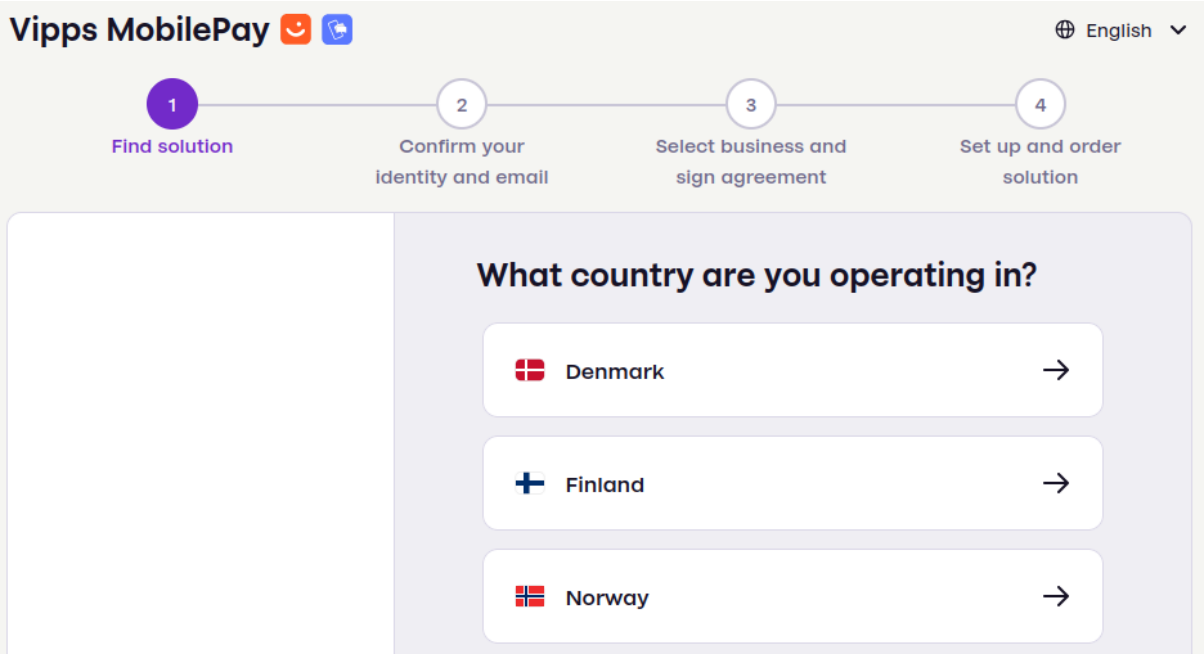
Source: Vipps MobilePay[100]

The API keys are available in the merchant portal. In order to gather information about the actual details required for accessing the API keys, the official sign-up process to create a new Vipps MobilePay merchant account was followed and the findings are summarised below. The merchant should first select the country that it is operating in, among Denmark, Finland and Norway.

[99] A partner is an entity that helps a merchant with creating integrations and managing their sales units

[100] https://developer.vippsmobilepay.com/docs/partner/partner-keys/

Figure 17: Vipps sign-up form – country selection



Source: Vipps MobilePay[101]

Then the merchant should verify his/her identity using a digital national ID or bank ID depending on the nationality country. The only options available as nationality are: Denmark, Finland, Norway and Sweden. Note that Swedish users must be invited by a merchant from Denmark, Finland, or Norway to be able to log in to the portal. As the JRC had not such an ID in disposal it was not possible to move forward with the account creation procedure. Nevertheless, in the above figure it is demonstrated that in the next step, the merchant selects the type of business and signs the agreement.

The requirements to use the Vipps MobilePay APIs are[102]:

- The guidelines of the Vipps MobilePay technical documentation should be followed.
- A Vipps MobilePay partner should follow the Vipps MobilePay Partner terms and conditions[103].

Note also that the Terms and Conditions for Vipps MobilePay Business solutions are available.[104]

[101] https://portal.vippsmobilepay.com/recommendation

[102] https://developer.vippsmobilepay.com/docs/terms-of-use/

[103] https://developer.vippsmobilepay.com/docs/partner/partner-terms/

[104] https://vippsmobilepay.com/en/legal/terms-and-conditions

4. Blockchain technology in online payments

4.1. Introduction

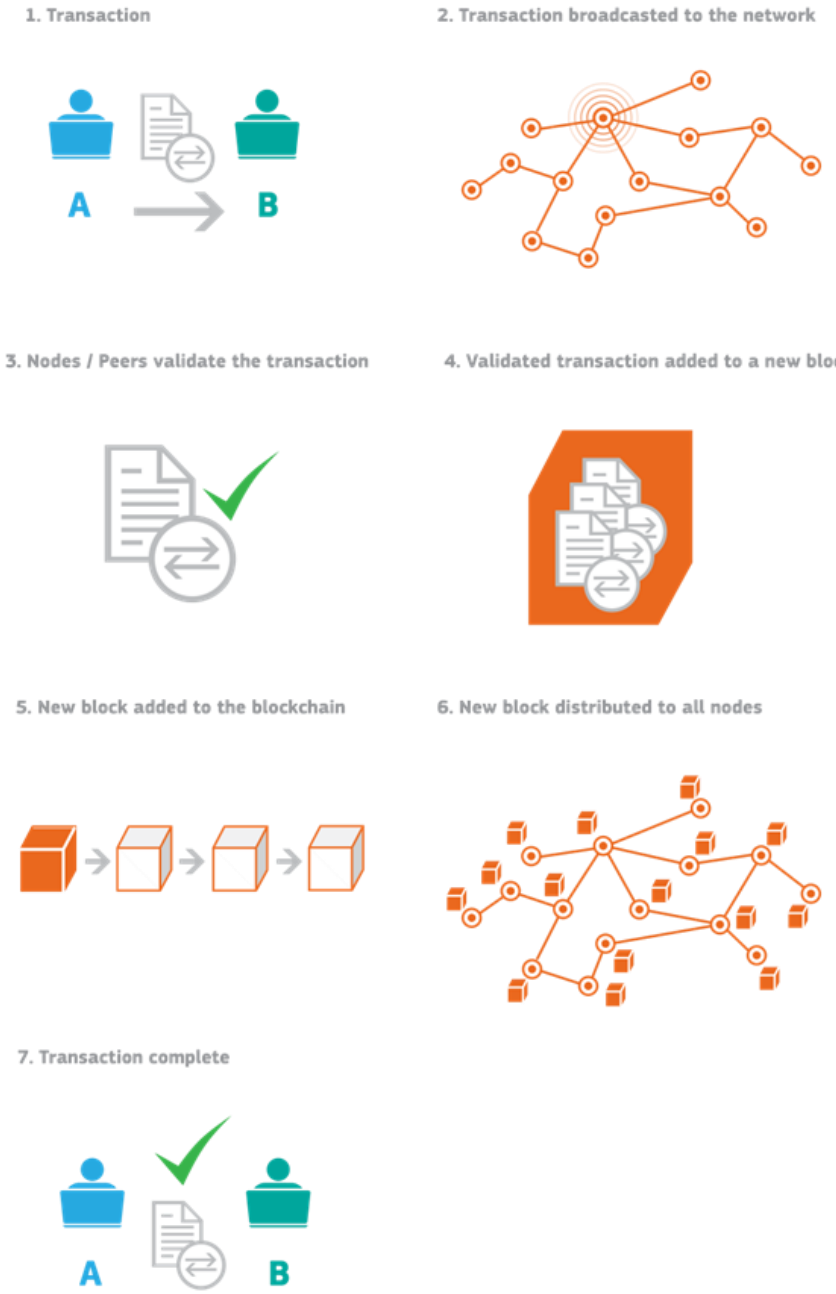
Blockchain and other Distributed Ledger Technologies (DLTs) are technologies enabling parties who have no particular trust in each other to exchange any type of digital data on a peer-to-peer basis with fewer to non-existent intermediaries. Data could represent any type of transaction or asset that can be translated in a digital form, such as money, insurance policies, contracts, land titles, etc.

To be clear in the terminology, blockchain is part of the broader family of DLTs. DLTs are particular types of databases in which data is recorded, shared and synchronised across a distributed network of computers or participants. Blockchain technology is a subset of DLTs that employs cryptographic techniques to record and synchronize data in 'chains of blocks'. The difference is about the way data is distributed, verified and registered. In short, all types of blockchain are DLTs but not all DLTs are blockchains. For the sake of simplicity this report will mostly use the term 'blockchain' or 'blockchains', but distinctions regarding DLTs will be made when necessary.

Blockchain is a database (ledger) operating in a distributed network of multiple nodes or computers that keeps track of data transactions [1]. It's called a 'blockchain' because of the particular way transactions are recorded and verified between parties. A transaction is requested by party A to party B. This transaction is broadcasted to a distributed network of nodes or computers, which will validate the transaction according to an agreed set of rules (a 'consensus' mechanism). When validated, this transaction will be bundled with others into a new 'block' and added to the blockchain.

The whole process ensures that each block is created in a way that irrefutably links it to the previous one and to the next one, thus forming a 'chain of blocks' or 'blockchain'. The unique record that forms a blockchain is shared by each node or computer in the network and is constantly updated and synchronized. In the end a blockchain stores the records of all transactions ever executed across a network. It is worth mentioning that depending on the implementation, the confirmation time may be quite long compared to traditional payments. For example, a relatively safe confirmation time for Bitcoin is 6 blocks or approximately 60 minutes, while for Litecoin that has a faster block creation time, it would be around 15 minutes. In most cases, such technologies are not ideal for instant payments.

Figure 18: Sequence relating to the creation and validation of a transaction on blockchain



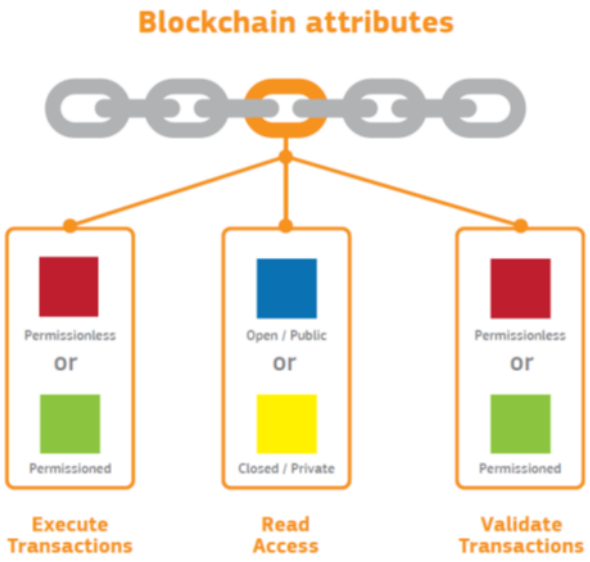
Source: Joint Research Centre own elaboration

There isn't just 'a blockchain' but many different blockchains with distinct functionalities and architectures. Blockchains can be distinguished depending on who can read, execute, and validate transactions. When anyone can read and access a blockchain it is categorised as 'public' or 'open'. That means that anyone can access a whole blockchain and read its contents. When only authorised entities have access, then a blockchain is considered 'closed' or 'private'.

Blockchains can be further categorised as 'permissionless' or 'permissioned' depending on who can send transactions and who can validate them. If anyone can send and validate transactions, the blockchain is called 'permissionless'. If entities need to be authorised to execute transactions,

validate transactions, or both, the blockchain is called 'permissioned'. At this point in their ongoing technical development, you can also find hybrid blockchains combining different aspects along a continuum [2] [3].

Figure 19: Different typologies of blockchain networks



Source: Joint Research Centre own elaboration

4.2. Stablecoins

Stablecoins are a particular type of cryptocurrency, which was created to decrease the price volatility of traditional cryptocurrencies. Stablecoins' value is pegged to another asset in order to maintain a stable price over time. Such assets could be for example, fiat currency or gold. The use of stablecoins has increased dramatically the last years; it is estimated that the stablecoin market has grown to $150 billion from just $15 million in 2017[105].

Apart from the difference on how their price is set, the technology and use of stablecoins is very similar to "traditional" cryptocurrencies. However, the fact that their price remains stable makes them a more reliable choice for specific use cases, such as remittances, B2B transactions, digital payments in developing countries, and international trade[106]. The largest stablecoins, by means of total market cap, currently are Tether, USD Coin, and DAI.

[105] https://www.forbes.com/sites/digital-assets/2024/03/29/stablecoins-what-are-they-used-for/

[106] https://www.europarl.europa.eu/thinktank/en/document/EPRS_BRI(2021)698803

4.3. Credit/Debit cards backed by cryptoassets

Crypto-backed credit and debit cards started to be offered on the market as a way to bridge the gap between the cryptocurrency world and the everyday real-world economy, allowing individuals who own cryptoassets to use them for everyday transactions.

In most of the cases, they are issued by fintech companies or cryptocurrencies exchanges in partnerships with two of the major card networks: Mastercard and VISA[107].

These cards can be either physical or virtual, and work like any other traditional credit or debit card: in case of a physical card it is possible to make a purchase at a retailer by inserting the chip into the POS or contactless thanks to the integrated RFID circuit. Virtual cards can instead be used for physical purchases if included in smartphone NFC payment systems like Apple Pay or Google Pay. Both types of cards can also be used for online payments in the same way of a traditional card.

At the time of this writing, three different types of crypto cards can be identified:

- Crypto-backed debit cards: they work by linking directly to a digital cryptoasset wallet. Whenever a purchase is made at a POS, the relevant amount of cryptocurrency is converted to fiat currency and withdrawn from the user account.
- Crypto-backed credit cards: which operate similarly to traditional credit cards, where the card issuer borrows a sum of money subject to a plafond to the card holder. In the case of crypto-backed credit card, the cryptoassets deposited into the card holder account are used as collateral, and the borrowed amount of money is usually subject to an interest rate.
- Crypto-backed prepaid cards: where users load specific amounts of cryptocurrency, which are immediately converted to fiat currency, to their balances.

In all cases, the spread in the cryptocurrency-fiat exchange rate, together with the various fees set from time to time by the various card issuers, form their revenue.

In addition, the crypto-backed card issuers typically implement various loyalty programs to incentivise card holders to use their cards and, moreover, to maintain a certain amount of cryptoassets frozen into the platform. As an example, several platforms, which implement their own native token, set various level of cashback based on the amount (i.e. total value) of native tokens held by the card holder, thus incentivising the card holder to hold (and keep locked) more and more tokens in order to obtain more benefits. Also, the fees structure of the card might be subject to the same mechanism, with lower fees granted to those card holders who hold the highest amount of native tokens.

It is also important to note that, while crypto-backed card issuers are not traditional banks, nowadays, at least in Europe, the major ones do follow all KYC and AML requirements, although a few ones still claim to offer non-KYC cards.

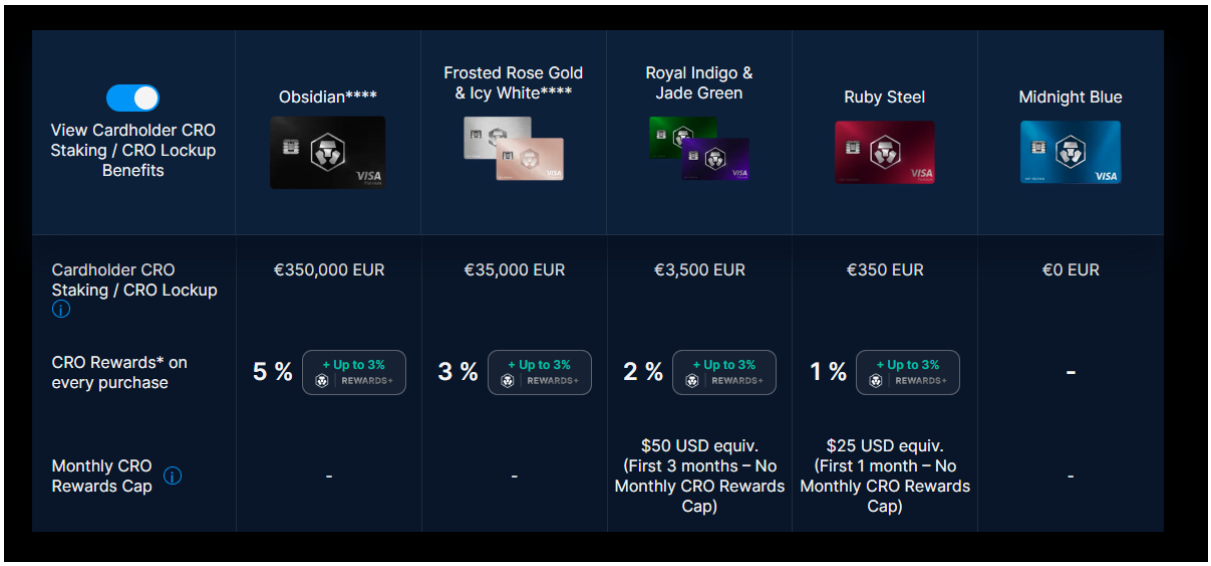
[107] American Express announced its first crypto-rewards credit card in partnership with Abra in 2022, but we are unable to find updated information on this, leading us to assume that such card is not currently issued: https://www.americanexpress.com/en-us/newsroom/articles/products-and-services/abra-announces-abra-crypto-card-on-the-american-express-netw.html

There are several crypto-backed cards available in the European market at the moment. In the following sections we will briefly present the main ones.

4.3.1. Crypto.com VISA Debit card

Crypto.com is one of the biggest cryptocurrency trading platforms and is active also in Europe (legal HQ is in Malta). Within the services and products available for their clients, a VISA debit card, available in 5 different tiers stands out.

Figure 20: Available tiers of the Crypto.com VISA debit card



Source: Crypto.com[108]

The 5 tiers differ in terms of benefit given to the cardholder, while they require, with the exception of the first tier, to stake[109] a certain amount of CRO (Cronos, the native cryptotoken of Crypto.com public blockchain). In exchange, the different tiers of this card recognise a certain percentage of cashback to the cardholder (subject also to monthly limit), paid out in CRO. Some other benefits, like the reimbursement of service subscriptions (e.g. Netflix or Spotify) are also available. Each card is also subject, among others, to top-up, withdrawal and inactivity fees[110]. There is also a fee to close your account.

4.3.2. Nexo CARD

The Nexo Card is a crypto-backed card that can be used both as a credit and as a debit card. In the credit mode, the card issuer uses the cryptocurrencies deposited into the card holder account as a

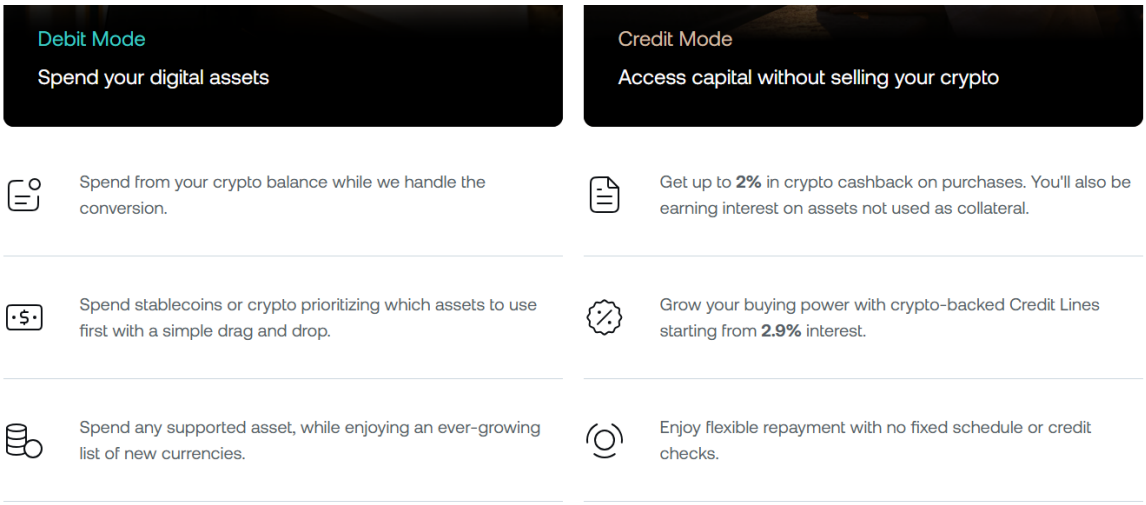
[108] https://crypto.com/it/cards

[109] Staking cryptocurrency is the practice of agreeing not to trade or sell a certain amount of digital token in exchange for something. In this case it means that the cardholder must possess in their crypto.com account a specific amount of CRO tokens

[110] https://help.crypto.com/en/articles/5977463-crypto-com-visa-card-fees-and-limits-europe-applicable-to-europe-residential-address-users-only#h_2e8f167ae5

collateral for the borrowed sums, which are lent with an interest rate that starts at a minimum of 2,9%. In the debit mode, instead, the card holder can decide the order of preferences in which the card issue will convert the various cryptocurrencies and stablecoins.

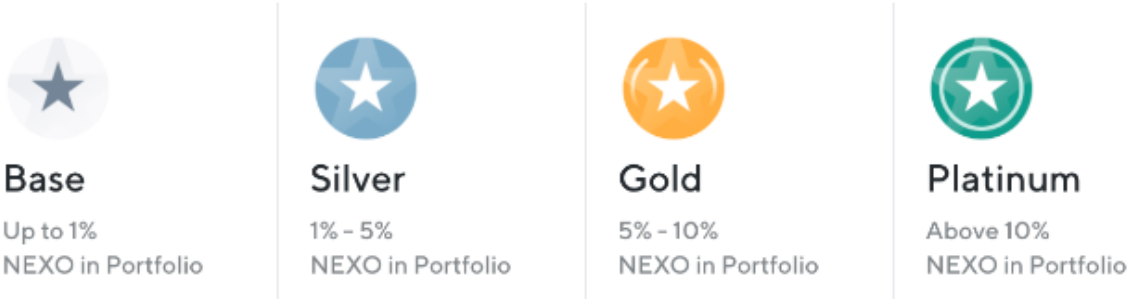
Figure 21: Nexo card: features of the credit and debit modes



Source: Nexo.com[111]

In credit mode, the cardholder is recognised a certain percentage of cashback on its purchases paid out in NEXO token[112]. This percentage is determined through the card holder loyalty tier[113], which is, in turn, decided by the percentage of NEXO token hold against the total of the cardholder portfolio.

Figure 22: Nexo card: requirements to access the various loyalty tiers



Source: Nexo.com[113]

[111] https://nexo.com/crypto-card

[112] NEXO is the native token of the NEXO platform (https://support.nexo.com/article/nexo-token-faq). It operates on the Ethereum blockchain network (through the ERC-20 standard) and on the BNB Beacon Chain Network through the BEP2 token standard. Please note that this second network is currently deprecated and is being migrated to the BNB Smart Chain network (BSC): https://www.bnbchain.org/en/bnb-chain-fusion

[113] https://support.nexo.com/article/nexo-loyalty-program-explained

4.4. Gift card – indirect blockchain payments

While the direct acceptance of cryptocurrencies as a mean of payment is not widespread (especially on large e-commerce), it is often possible to still perform an indirect payment in cryptocurrency to buy physical goods or services. This is done through gift cards.

A gift card is a type of prepaid card or digital voucher that can be used to purchase goods or service from a specific retailer (either online or at a physical store). The face value is therefore paid in advance with respect to the actual purchase, and each retailer specifies the specific terms and conditions[114] of its own gift cards. Several, if not the majority of retailer businesses[115] offers this payment option, that is usually used, as the name suggest, as a gift for celebrations, or as a benefit for employees due to the preferential tax regime (for both the company and the employee) compared to ordinary salary.

There are several intermediaries which allow the purchase of hundreds of different gift cards (with various face values) paying with cryptocurrencies.

Some of these include:

- Bitpay[116]
- Coingate[117]
- Bitrefill[118]

Purchasing a gift card on these websites is fairly easy and most of the time does not require having an account or leaving personal data other than an email address to receive the gift card code. As an example, we will exemplify the purchase of an Amazon.it gift card on Bitrefill.

Bitrefill offers several gift cards for purchase with cryptocurrencies. It is possible to select the specific country of interest to see a targeted list of cards available in that specific country.

[114] e.g. deadline to spend the funds, the need to use the face value of the card is a single transaction or the possibility of making multiple purchases, any exceptions to the type of good that can be purchased, etc.

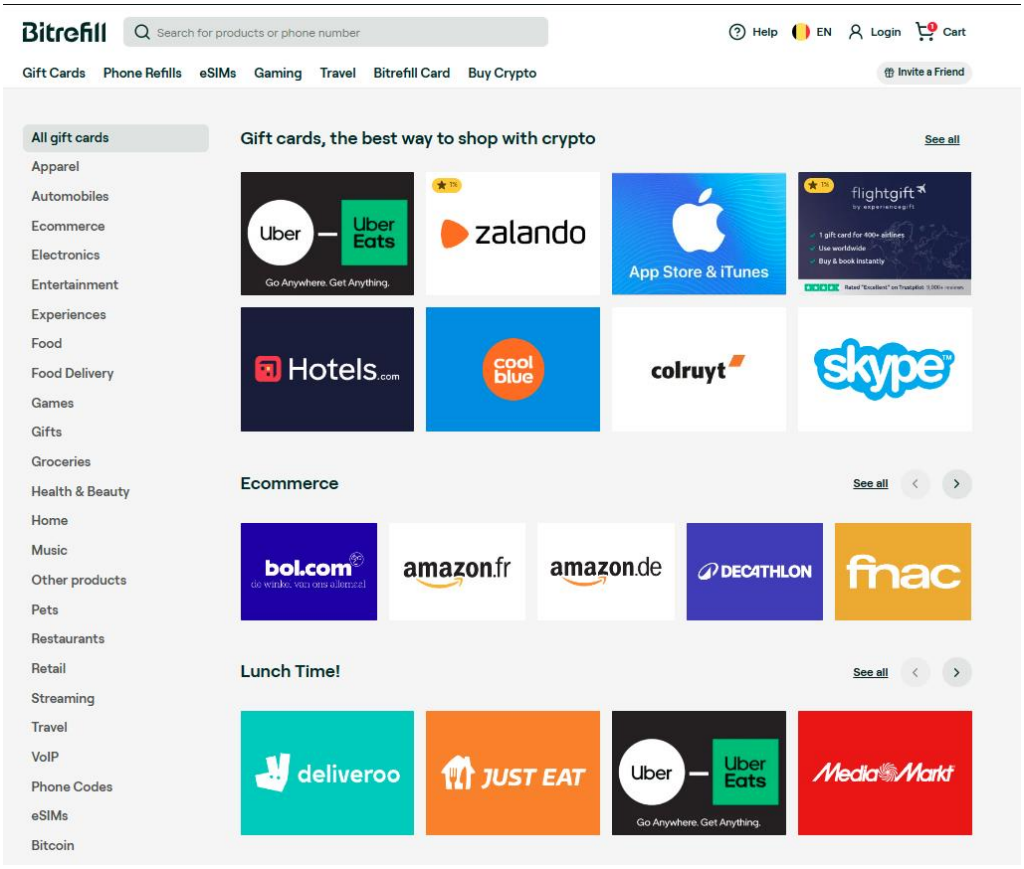
[115] all retail sectors are virtually included, including large e-commerce (e.g. Amazon), tourism services (e.g. Airbnb), grocery shops (e.g. Lidl), petrol stations (Esso) and transportation companies (e.g. Trenitalia)

[116] https://bitpay.com/gift-cards/

[117] https://coingate.com/gift-cards/category/all

[118] https://www.bitrefill.com/it/en/

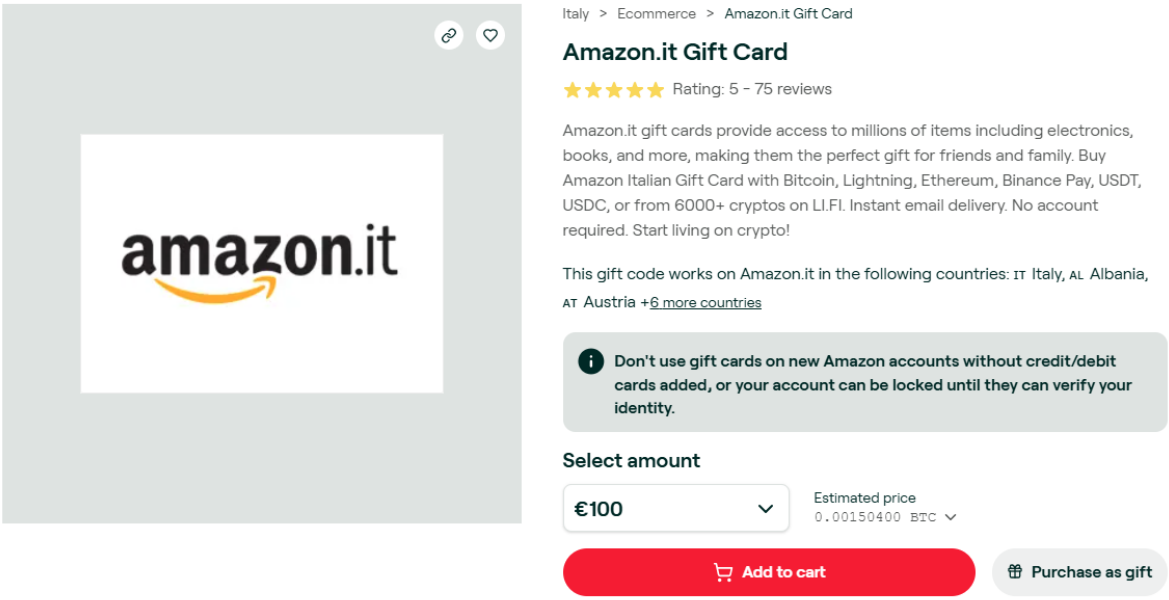
Figure 23: Bitrefill.com – example of gift cards available for purchase



Source: Bitrefill.com[118]

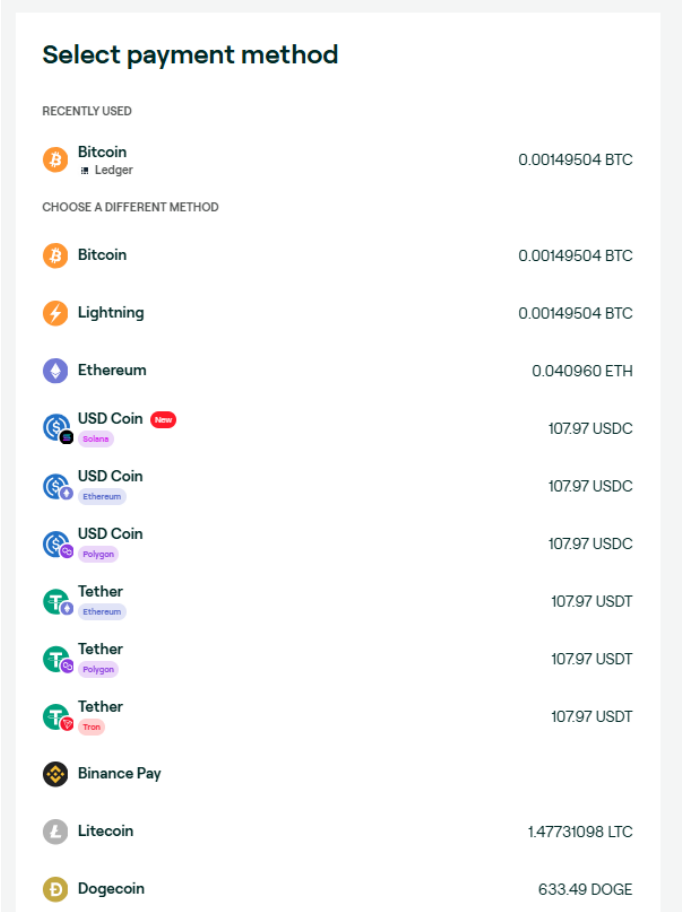
Buying a specific card is rather easy. It is sufficient to select the card, specify the desired face value and select in which supported cryptocurrency we would like to pay.

Figure 24: Bitrefill.com – Purchase of a 100 euros gift card for Amazon.it



Source: Bitrefill.com[118]

Figure 25: Bitrefill.com – Purchase of a 100 euros gift card for Amazon.it – Accepted crypto-payments methods

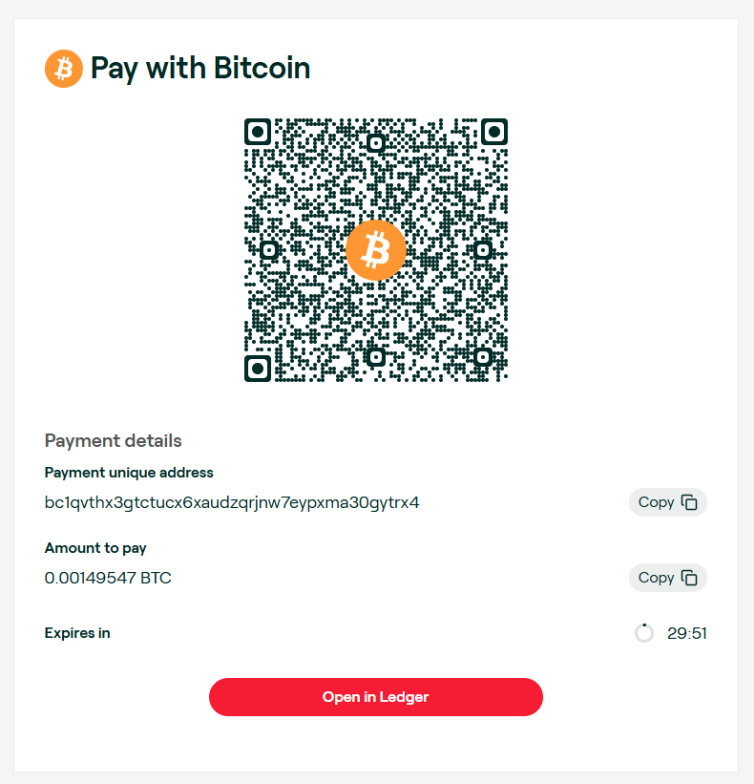


Source: Bitrefill.com[118]

Taking as an example the purchase of a 100 Euro Amazon.it gift card in Bitcoin, we are shown the exact amount in Bitcoin to pay and the blockchain address towards which we need to send the payment. The payment can be done using whatever wallet we prefer, as the only needed information are indeed the blockchain address and the amount. As the figure shows us, the system allows us 25 minutes to perform the transaction.

Once we insert the transaction, even if it is not yet confirmed in the Bitcoin blockchain, Bitrefil recognise we have inserted the transaction and notify us it is waiting for its confirmation (both on the webpage and via email). At this point we can also close the webpage as the confirmation might take a long time depending on the transaction fee we decided to pay and the Bitcoin network congestion. Once confirmed in a block, we will get an email with a link to a page where we are able to see the gift card code, which we can redeem directly on the Amazon website.

Figure 26: Bitrefill.com – Purchase of a 100 euros gift card for Amazon.it – Payment popup showing transaction details



Source: Bitrefill.com[118]

Although surely not so straightforward as a credit card payment might be, this indirect payment way enables the actual use of cryptocurrencies in virtually all the big brands, physical stores and e-

commerce available. This solution is also more attractive to cryptocurrency holders considering that gift cards purchases is less scrutinised by fiscal authorities compared to credit card payments[119].

4.5. Crypto payment gateways

Crypto payment gateways are services that enable merchants to accept transactions in various cryptocurrencies. Like traditional payment gateways used for credit card processing, crypto payment gateways facilitate the transaction process by ensuring the secure transfer of payments. These services take away the need for the merchant to maintain multiple wallets for multiple cryptocurrencies and having to monitor the status of each blockchain for confirmation. Often, they also offer the option to convert directly the received payment from cryptocurrency to fiat currency. In the context of crypto payments all the services mentioned here act both as payment gateways as well as payment processors.

A typical workflow of a gateway is the following:

— The merchant creates an account with the payment gateway that provides them with a custodial digital wallet. The merchant decides on the list of cryptocurrencies they want to receive payment in.

— Typically, the merchant will integrate with the gateway on their website either through the API, prebuilt plugins for ecommerce platforms or quick payment buttons. Some of the providers offer Point of Sale mobile apps for in person payments.

— When a customer decides to make a purchase and chooses to pay with cryptocurrency, they select the desired digital currency at checkout.

— The payment gateway generates a unique transaction by providing the customer with the merchant's wallet address and/or a QR code corresponding to the payment amount and currency.

— The customer transfers the cryptocurrency funds through their own digital wallet.

— Once funds are received and the transaction is considered finalised (depending on the blockchain system that can be one or more blocks, in Bitcoin typically 6) the payment processor credits funds to the merchant’s wallet and the merchant is notified either through the API or email so they can finalise the transaction with their customer.

Here is a non-exhaustive list of the largest crypto payment gateways based or active in Europe:

— Coingate[120] – Based in Lithuania.

— CoinPayments[121] – Based in Lithuania.

[119] Article 19 of EU Regulation 2024/1624 exempts obliged entities from applying customer due diligence for payment instruments not reloadable that hold an amount up to 150 euros and which are used exclusively to purchase goods or services provided by the issuer, or within a network of service providers

[120] https://coingate.com

[121] https://www.coinpayments.net/

— Nowpayments.io[122] – Incorporated in Seychelles but have offices in the Netherlands.

— Lunu.io[123] – Based in Lithuania.

— Bitpay [124] – They claim to be the biggest crypto payment processor in the world.[125] Based in the U.S. with offices in the EU.

— Binance Pay[126] – The cryptocurrency payment gateway arm of Binance, a very popular crypto exchange and custodial wallets service. Based in the U.S.

— Crypto.com[127] – provides crypto payment services as well as standard crypto exchange services. Based in the US.

— Confirmo[128] – Based in Slovakia.

4.5.1. Bitpay

Bitpay[129] is one of the oldest cryptocurrency payment gateways. It was established in 2011. It provides services for individuals as well as businesses. Apart from crypto payment gateway services it also provides a multiplatform non-custodial[130] digital wallet as well as a crypto backed credit card.

Bitpay supports two main payment types: online and in store payments.

[122] https://nowpayments.io

[123] https://lunu.io/

[124] https://bitpay.com

[125] https://bitpay.com/blog/content/files/2024/04/2024-bitpay-crypto-spending-report.pdf

[126] https://pay.binance.com/en

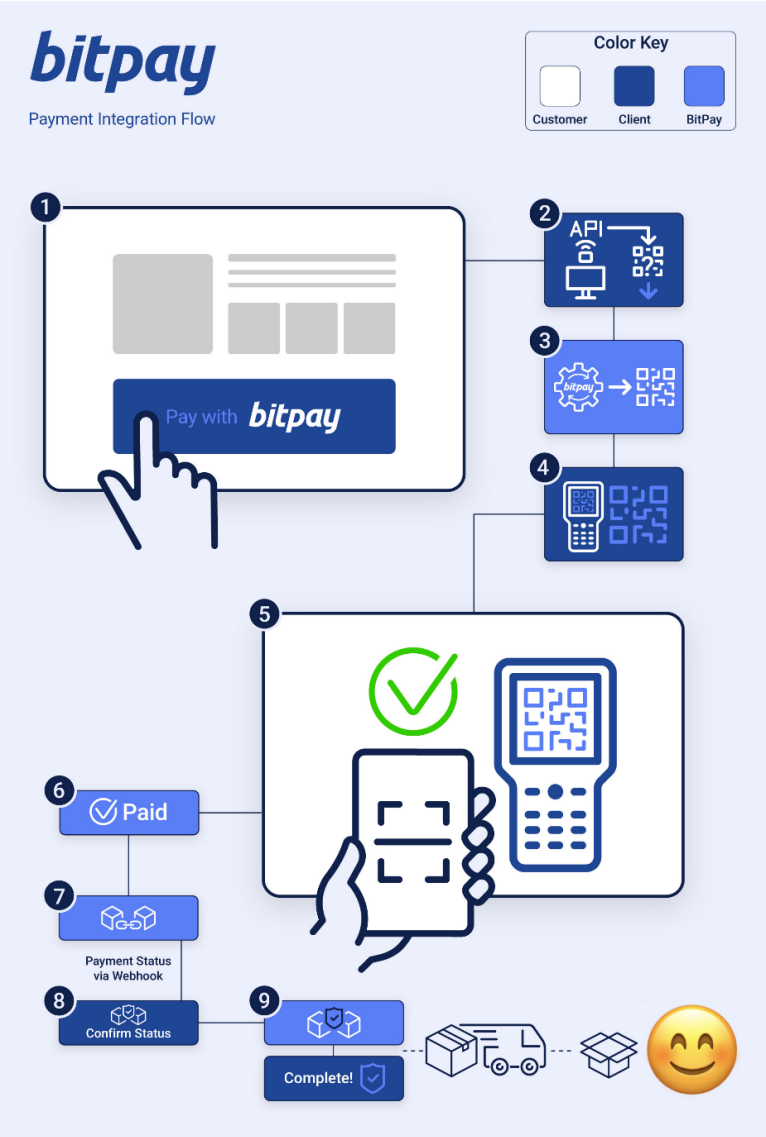
[127] https://crypto.com/

[128] https://confirmo.net/

[129] https://www.bitpay.com/

[130] A "non-custodial" digital wallet means that the cryptographic key necessary to sign the transaction (and therefore to actually own the cryptocurrencies assets) is not stored in the wallet providers servers but only in the specific device where the wallet is installed (e.g. a mobile phone). Users need to manually backup a recovery phrase or they risk losing access to their assets

Figure 27: Bitpay: Payment Integration Flow



Source: bitpay.com[131]

1. When the customer reaches checkout (online or in person) they select "Pay with Bitpay."
2. The merchant platform calls the Bitpay API to generate a new payment.
3. Bitpay creates a payment with a "New" status. An associated URL is created and sent back to the merchant via an HTTP response.
4. The merchant redirects the customer to the URL provided by Bitpay, where the customer can scan a QR code to enter the payment process (in store, the merchant presents the QR code to the customer.) This process includes selecting their wallet provider and payment currency, then authorizing the transaction.

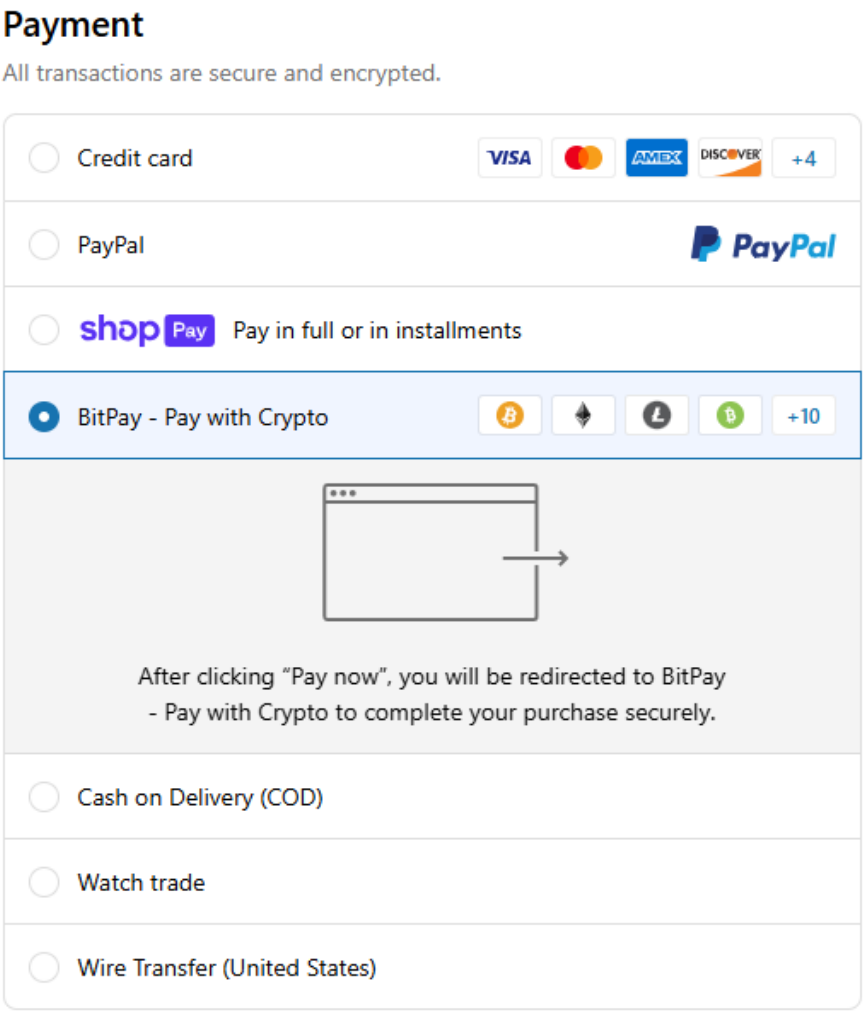
[131] https://developer.bitpay.com/docs/checkout-integration

5. The customer will see confirmation on their device that the payment was sent and will be redirected back to the merchant's site if applicable.
6. On the Bitpay end, the payment will now be changed to a status of "Paid."
7. As the transaction processes on the blockchain, Bitpay will transmit the latest payment status via Webhook to the merchant.
8. When the merchant receives the webhook indicating the payment is "Complete," the merchant should call the Bitpay API again to confirm this status. This is recommended because Bitpay's web hooks are unsigned, and therefore should not be trusted outright.
9. If the API call confirms the payment "complete" status, the funds have been applied to the merchant ledger, and the order can be fulfilled.

From the buyer point of view, deciding to paying via Bitpay is somehow comparable with the same user experience they have when paying with a traditional payment gateway like PayPal.

On the checkout page, the option to use Bitpay as payment method will be shown together with the other payment options, like in the example below.

Figure 28: example of website integrating payment through BitPay



Source: investmentwatches.com[132]

As an alternative, the following button could be present on the page to notify this specific payment option

Figure 29: Button usually shown on webpage to signal that payments through Bitpay are accepted

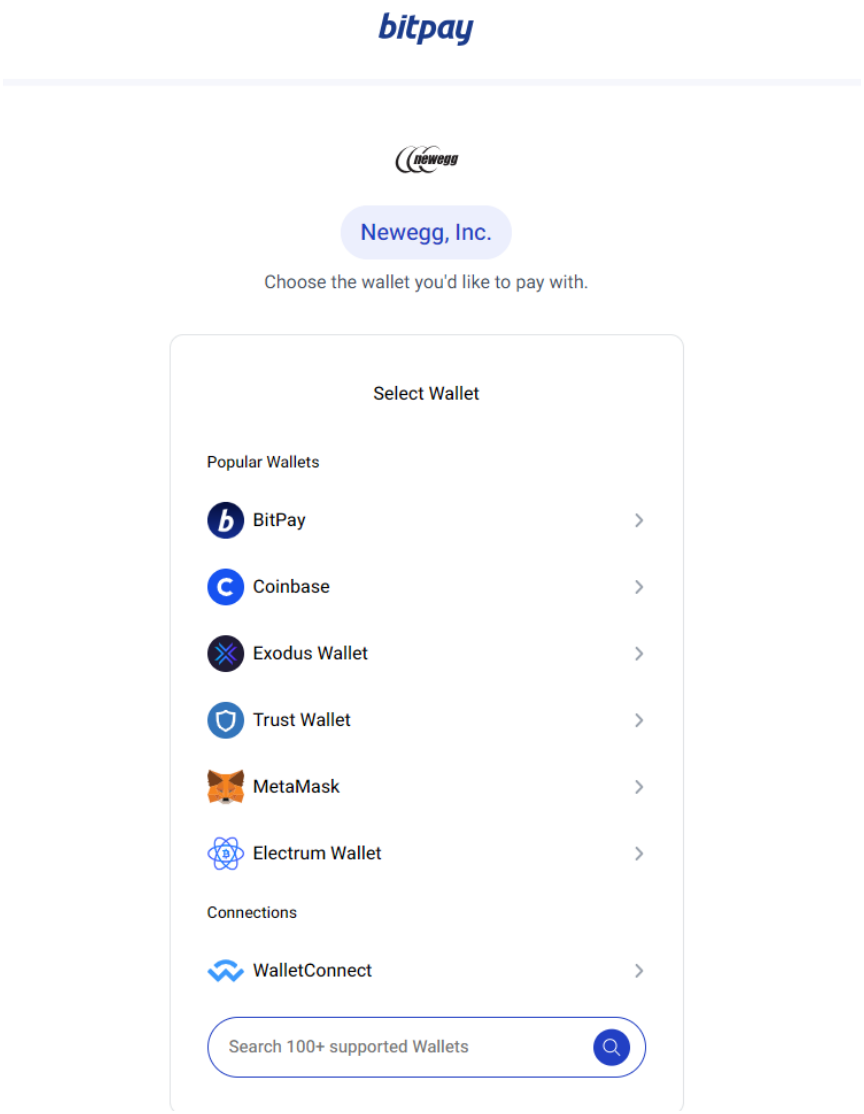
Source: bitpay.com[133]

[132] https://www.investmentwatches.com

[133] https://brand.bitpay.com/7b6hnbyxw/p/106792-brand-requirements/b/60db79

When selecting to pay with Bitpay, a separate page will prompt the buyer to select from which wallet they would like to create, sign and broadcast the payment transaction.

Figure 30: Bitpay payment popup: wallet selection



Source: bitpay.com – payment simulation

The desired wallet application will open up, from which the buyer will be able to check the transaction details and confirm the payment.

4.5.2. BTCPay Server

BTCPay Server[134] is a free and open-sourced (MIT License) self-hosted bitcoin payment gateway that allows anyone (individuals and businesses) to accept bitcoin payments without relying on an intermediary and without any fee[135].

Simple to use, it shows a pop-up to the buyer with the necessary data to perform the bitcoin transaction (i.e. BTC amount and recipient address), and a convenient QR code that might be scanned using a Bitcoin Wallet app to automatically configure the transaction detail.

Figure 31: BTCPay Server payment window popup – QRCode with transaction details

Source: Own simulation using a local instance of BTCPay Server

Once the transaction has been broadcasted to the blockchain, BTCPay is able to automatically verify its presence and confirm to the buyer the correct execution of the payment. Considering the nature of the bitcoin blockchain, the buyer would nevertheless have to wait for the transaction to be inserted in a valid blockchain block. Bitcoin payments are for this reason considered asynchronous: the payment confirmation will be sent to the buyer at that point only.

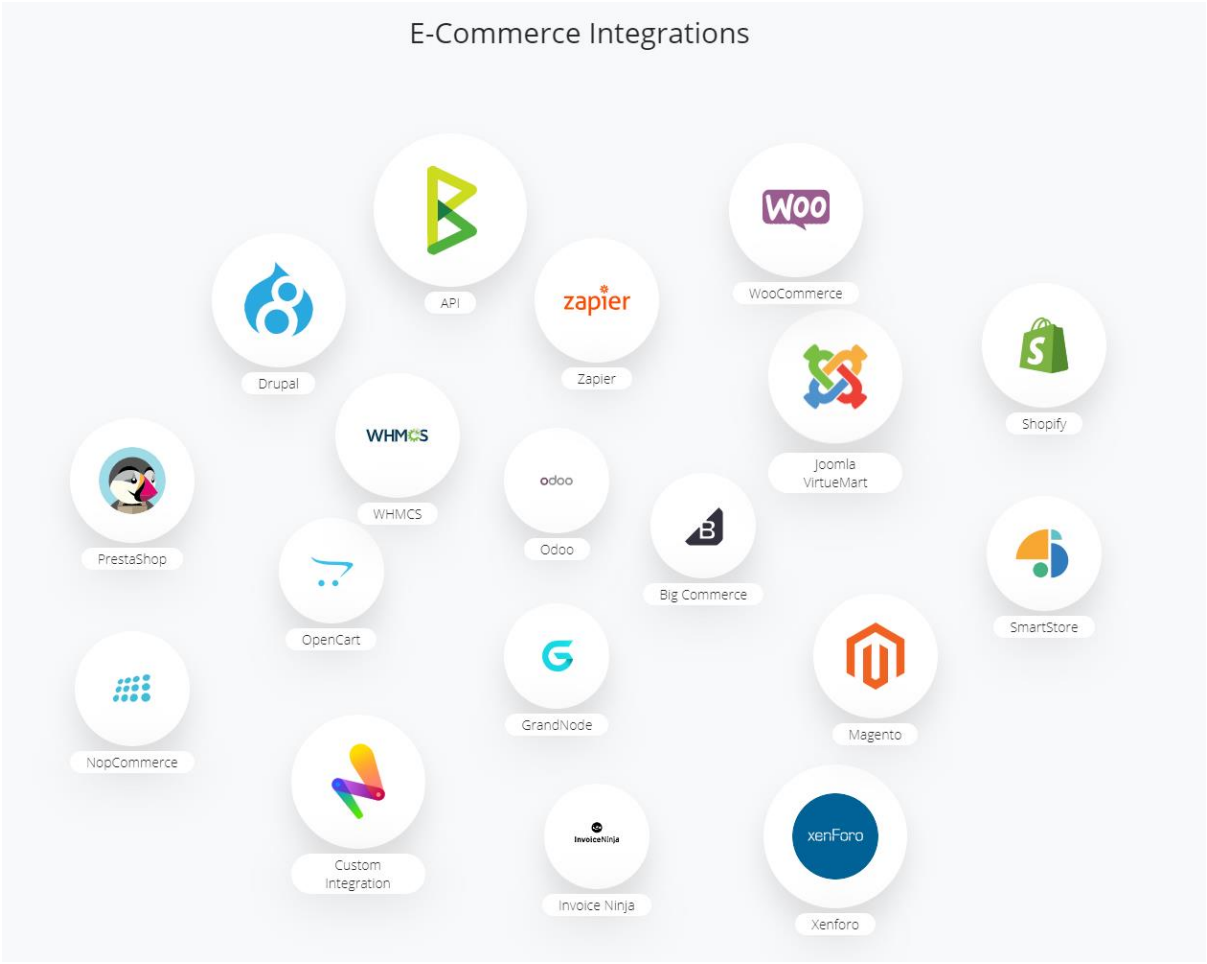
[134] https://btcpayserver.org/

[135] Both the merchant and the buyer are not subject to any fee other than the network transaction fee, which is native to bitcoin and it is paid by the buyer when submitting the transaction

BTCPay Server is a separate component from the website that hosts the merchant e-commerce, and it is accessed via the use of APIs.

Several free and open-sourced plugins exist for the major CMS (Content Management Systems) used nowadays to build websites and e-commerce.

Figure 32: BTCPay server e-Commerce Integrations



Source: btcpayserver.org[136]

As explained in the project's documentation[137], a BTCPay Server instance can be easily installed on a multitude of devices, including low-powered ones like Rasperry Pi, and the common setup to start accepting bitcoin payments requires the following component:

— *Nginx*, a web server acting as a reverse proxy to receive web requests for interacting with BTCPay,

— *BTCPay Server*, which is the payment processor,

[136] https://btcpayserver.org

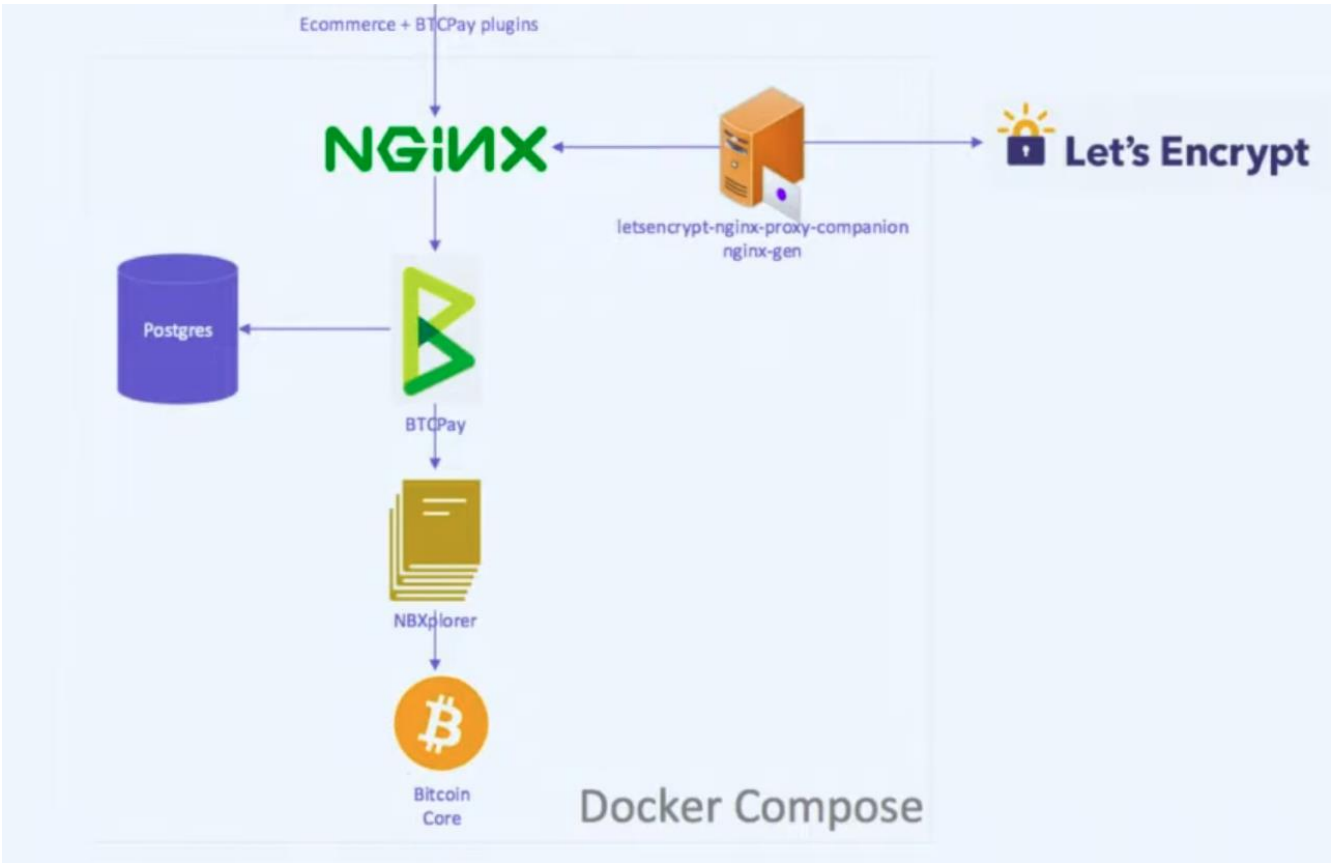
[137] https://docs.btcpayserver.org/Development/

— *NBXplorer*, a lightweight block explorer, used for tracking payments onto the blockchain,

— *Bitcoin Core*: the actual bitcoin node software, used to interact with the bitcoin blockchain and retrieve new blocks. It can work in pruned-mode[138] and therefore it does not need to store the entire blockchain,

— *PostgreSQL*, a database engine.

Figure 33: BTCPay Server instance – common architecture



Source: YouTube[139]

Nevertheless, in the case a merchant does not wish to setup its own instance of BTCPay Server, it is also possible to use a shared third-party instance of BTCPay Server, usually against a monthly fee.

[138] A bitcoin node working in pruned-node, discards the blockchain blocks after it has received and processed them, keeping only the set of Unspent Transaction Output (UTXO) that is what you need to verify the correctness of payment

[139] https://www.youtube.com/watch?v=Up0dvorzSNM&t=220s

4.5.3. Who accepts cryptopayments in the EU?

Although a comprehensive list of merchants directly accepting cryptocurrency payments in the EU is not available, online searches on cryptocurrencies portals and forums provide a non-exhaustive list, which is given here. Note that cryptocurrency acceptance may not be available from all Member States. This list is of course constantly evolving, both in terms of new merchants starting to accept such payments and vice versa. As mentioned above many more can be indirectly paid via gift cards.

— All crypto exchanges. A client can transfer funds from their wallet to the wallet of the crypto exchange in order to exchange it for other cryptocurrencies, fiat currency or pay for services rendered by the exchange.

— Air Baltic

— Tesla[140]

— Online retailers

  - Alza.cz - One of the largest online consumer electronics retailers in central Europe.
  - Shopinbit. Polish retailer. Sells electronics, consumer goods, acts as a travel agency, car rental agency.

— VPN services

  - NordVPN
  - ExpressVPN
  - AirVPN
  - Surfshark
  - Mulvado

— Email services

  - Proton mail

— Blockchain services

  - Chainanalysis

— Shopping aggregators

  - Spendabit.co

— Access to LLM APIs

  - aimlapi.com

[140] https://www.tesla.com/support/dogecoin

5. Conclusions

This study has provided an in-depth analysis of online payment systems in the EU, covering various payment methods, APIs, and SDKs used by online payment platforms, as well as the emerging role of blockchain technology in online payments.

The report has highlighted the diversity of online payment options available to EU citizens, including payment cards, mobile wallets, bank transfers, and instant payments, among others. Furthermore, it has examined the APIs and SDKs used by popular online payment platforms, such as Samsung Pay, Google Pay, Amazon Pay, Blik, Paypal and Vipps MobilePay and discussed their access rules and conditions.

The study has also explored the use of blockchain technology in online payments, including stablecoins, credit/debit cards backed by cryptoassets, and crypto payment gateways, and has identified their potential benefits and limitations.

Overall, this report aims to contribute to a better understanding of the online payment landscape in the EU and to inform policy discussions on the development of online payment systems

References

[1] Wright, A. and De Filippi, P. 2015. 'Decentralized Blockchain Technology and the Rise of Lex Cryptographia', SSRN / Social Science Research Network.

[2] Peters, G.W., and Panayi, E. 2016. 'Understanding Modern Banking Ledgers Through Blockchain Technologies: Future of Transaction Processing and Smart Contracts on the Internet of Money', in: Banking Beyond Banks and Money, ed. by P. Tasca T. Aste, L. Pelizzon and N. Perony, Springer, Heidelberg, pp. 239–78. http://dx.doi.org/10.1007/978-3-319-42448-4

[3] Danezis, G. and Meiklejohn, S. 2015. 'Centrally Banked Cryptocurrencies'. http://dx.doi.org/10.14722/ndss.2016.23187

List of abbreviations and definitions

Abbreviations	Definitions
API	Application Programming Interface
BNPL	Buy Now Pay Later
COD	Cash on Deliveryt
CVV/CVC	Card Verification Value / Card Verification Code
EMV	Europay, Mastercard and Visa
eSE	Embedded Secure Element
HCE	Host Card Emulation
IBAN	International Bank Account Number
NDA	Non-Disclosure Agreement
NFC	Near Field Communication
PG	Payment Gateway
PIS	Payment Initiation Service
PSD2	Payment Service Directive 2
RBA	Risk Based Authentication
SCT	SEPA Credit Transfer
SCT Inst	SEPA Instant Credit Transfer
SDK	Software Development Kit
SE	Secure Element
SEPA	Single Euro Payments Area
TEE	Trusted Execution Environment
UCM	Universal Certificate Management

List of figures

List of tables

Getting in touch with the EU

In person

All over the European Union there are hundreds of Europe Direct centres. You can find the address of the centre nearest you online (european-union.europa.eu/contact-eu/meet-us_en).

On the phone or in writing

Europe Direct is a service that answers your questions about the European Union. You can contact this service:

— by freephone: 00 800 6 7 8 9 10 11 (certain operators may charge for these calls),

— at the following standard number: +32 22999696,

— via the following form: european-union.europa.eu/contact-eu/write-us_en.

Finding information about the EU

Online

Information about the European Union in all the official languages of the EU is available on the Europa website (european-union.europa.eu).

EU publications

You can view or order EU publications at op.europa.eu/en/publications. Multiple copies of free publications can be obtained by contacting Europe Direct or your local documentation centre (european-union.europa.eu/contact-eu/meet-us_en).

EU law and related documents

For access to legal information from the EU, including all EU law since 1951 in all the official language versions, go to EUR-Lex (eur-lex.europa.eu).

EU open data

The portal data.europa.eu provides access to open datasets from the EU institutions, bodies and agencies. These can be downloaded and reused for free, for both commercial and non-commercial purposes. The portal also provides access to a wealth of datasets from European countries.

Science for policy

The Joint Research Centre (JRC) provides independent, evidence-based knowledge and science, supporting EU policies to positively impact society

EU Science Hub
Joint-research-centre.ec.europa.eu

Publications Office
of the European Union