ChatGPT

# Vipps MobilePay Integration for Averdi’s Accounting Services

Vipps MobilePay is Norway’s leading mobile payment platform, offering APIs for online checkout, point-of-sale (POS) payments, recurring subscriptions, donations, and more [1] [2]. The Vipps API platform uses a **unified framework** – all products share the same authentication and message formats – so combining multiple payment flows is straightforward [1] [2]. For example, Vipps provides an **ePayment API** for online or in-store transactions (cards, Vipps, MobilePay), a **Checkout API** with pre-built plugins for Shopify, WooCommerce, etc. [3], a **Recurring API** for subscriptions and memberships [4], and a **Donations API** for fundraising campaigns [5]. These APIs, together with a **Report API** for fetching transactions and an SDK, give developers a complete toolkit to integrate Vipps. Because Vipps centralizes keys and uses consistent endpoints, we can treat all these functions in one system [1] [2], but implementing them still requires careful development. For Averdi’s firm, this means hiring a dedicated developer to set up and maintain these integrations with clients’ accounting systems.

*Vipps offers an all-in-one online checkout (Checkout API) with ready-made plugins for platforms like Shopify and WooCommerce [3], simplifying e-commerce integration.*

## E-Commerce Checkout and Payment Integration

For online stores, Vipps’s **Checkout API** provides a complete payment solution that handles the entire checkout flow (payment, shipping, customer info) [6] [3]. Vipps even offers *open-source plugins* (Shopify, WooCommerce, Adobe Commerce/Magento, Prestashop, etc.) so we can plug Vipps straight into a webshop without building from scratch [3]. By integrating this into our clients’ webshops, payments will automatically be processed via Vipps MobilePay. The developer will connect the checkout flow to Averdi’s accounting tools (e.g. via 24SevenOffice or PowerOffice APIs) so that each order generates an invoice and marks it paid when the Vipps transaction settles.

Beyond the standard Checkout, Vipps also provides an **ePayment API** to accept payments in various contexts. This API supports multiple user flows, including *online payments*, *in-store payments*, and *one-time QR-code payments* [7]. For example, the business could generate a unique Vipps QR code or initiate a POS payment (Vipps app scanning) for each sale. The integration will capture these payment events and forward them to the accounting system. In practice, the developer would call the ePayment endpoints to create payment requests (or use Shopify/Woo/WooCommerce plugins) and then use the **Report API** to retrieve transaction results for reconciliation. All of this requires code to match each Vipps transaction to the correct customer invoice.

## Recurring Payments and Donations

For clients that need recurring billing (membership fees, subscriptions, donations), Vipps’s **Recurring API** is ideal. It allows customers to set up automatic payments for subscriptions, membership dues, or installment plans without re-entering card details [4]. Customers get notified of withdrawals in their Vipps app, making it user-friendly. There are even ready plugins for recurring payments (e.g. WooCommerce Recurring Payments) that we can leverage [8]. Our system can use the Recurring API to create and manage these payment agreements, and automatically post each withdrawal to the accounting system as it occurs.

Vipps also offers a **Donations API** and built-in donation products for fundraising. Non-profits and clubs can create Vipps “collection” campaigns, so supporters pay directly in their Vipps app. The Donations API then lets us retrieve details of each donation after the fact [5]. While organizations don’t need to integrate the API just to use Vipps donations, we can use it to build a customized donation portal or dashboard. For example, the developer could create a module where a sports club shares a Vipps donation link or QR code at events, and every donation (one-time or recurring) is automatically logged in the club’s books.

*Vipps supports recurring payments for subscriptions and donations [4]. For instance, we can enable customers to “pay monthly” for a service or membership via Vipps.*

## Integration with Accounting Systems

A key goal is to **streamline bookkeeping**: every Vipps payment should post directly into 24SevenOffice or PowerOffice with minimal manual work. To do this, we will use each platform’s API together with Vipps’s data. Vipps provides a **Report API** for accounting partners that lets us retrieve all transaction and settlement data across clients (with special “accounting keys” [9]). As an accounting partner, we can use these keys to pull each client’s payments, fees, and payouts from Vipps into our system [9]. This ensures our software always has the latest Vipps data for reconciliation.

In 24SevenOffice, there is already an existing Vipps integration (by iizy) that automates posting the daily settlement report [10]. It reads Vipps’s report and creates accounting entries for customer payments, Vipps fees, and the net payout deposited to the bank [10]. We can mirror this approach: our modules will fetch the settlement report via Vipps’s Report API and use the 24SevenOffice API (or PowerOffice API) to book entries. Specifically, each day we would debit the Vipps intermediary account and credit the bank account when the payout arrives, while offsetting against customer invoices.

For **PowerOffice**, the developer should follow the recommended workflow from PowerOffice’s API documentation. PowerOffice treats payments as separate accounting events from invoices [11]. For example, when a customer pays a 12 500 NOK invoice with Vipps, the invoice is first posted (credit sales, debit customer), and then a separate payment event is posted (debit Vipps account *NOK 12 500*, credit customer ledger with a custom matching reference) [12]. The invoice is then matched and closed. When Vipps later pays out to the bank, a final entry debits the bank (1920) and credits the Vipps account (e.g. 1903) [13] [14]. In practice, we will batch all of a day’s payments into one **BankJournal** voucher: debit the Vipps account for the total and credit each individual customer’s ledger entry using a unique matching reference [14]. This automated BankJournal posting (debit Vipps, credit customer1,2,… etc) exactly follows PowerOffice’s guidelines [14]. By coding these steps against PowerOffice’s API, we ensure each Vipps payment reconciles correctly.

Both systems allow us to assign separate asset accounts for Vipps, credit card, and other channels [15]. For bookkeeping clarity, we’ll maintain a dedicated account (e.g. 1903) for Vipps transactions. Every payment is matched to its invoice via a unique reference code, so there are no open items. This approach reduces errors and manual work: once set up, the integration will automatically import Vipps transactions into the accounting ledger each night.

## Proposed Integration Modules and Services

We envision a **modular subscription service** (SaaS) that our firm can offer to clients. Key modules include:

- **E-Commerce Checkout Integration Module:** A plugin or connector for popular webshops (Shopify, WooCommerce, Magento) that adds Vipps as a checkout option. This module uses Vipps's Checkout or ePayment APIs to process sales and then calls the accounting API to create or mark invoices. By automating invoice creation and payment capture, it speeds up bookkeeping for online stores. (Leverages Vipps Checkout API and plugins [3] .)

- **Recurring Billing & Membership Module:** A system to manage subscriptions and memberships. Using Vipps Recurring API, this module can sign up members for automated payments (e.g. gym membership, club dues) [4] . Each recurring withdrawal is posted to the ledger as a payment against the corresponding invoice. This module could include customer management (through Vipps Login API) and send reminders. (Supports monthly/annual fees without manual invoicing each period.)

- **Donation & Event Payments Module:** A toolkit for volunteer organizations and clubs. It would generate Vipps collection campaigns (one-time or recurring) and QR codes for events. After each donation or ticket sale, the system records the payment, issues a receipt, and syncs the total to the accounting books. We could build a simple online portal where organizers register an event, get a Vipps payment link, and automatically receive a transaction report. (Uses Vipps Donations API and QR/pay flows [5] [7] .)

- **POS/QR Payment Module:** For small shops or events, this module handles in-person Vipps payments. It generates on-the-fly QR codes or connects to a POS device via Vipps ePayment API [7] . Every scan or tap result is captured and forwarded to accounting. This makes checkout cashier-free: the payment screen pops up on the customer's phone and we get the transaction detail via Vipps's API.

- **Reconciliation & Reporting Module:** This background service automatically retrieves daily transaction and settlement data from Vipps (via the Report API) for *all* clients. It formats the data and pushes it into PowerOffice or 24SevenOffice (through their APIs) to complete the entries for customer payments, fees, and bank deposits [10] [14] . We can also provide a simple dashboard so clients (or accountants) can review payment reports, fees, and outstanding items. This module acts as the integration engine, tying Vipps data to the accounting ledgers.

Each module would be offered as part of a subscription package. Clients pay a monthly fee for the software and maintenance. New features (e.g. additional webshops, updated APIs) can be rolled out seamlessly. By modularizing, we can tailor the solution: a small sports club might only use the Donation/Event and Reconciliation modules, while a retailer uses the E-Commerce and POS modules.

## Free Integration Guide for Sports Clubs

As part of our strategy, we will publish a **free Vipps integration guide** on our website for volunteer organizations (idrettslag, charities, clubs). This guide will explain, in layman's terms, how to use Vipps for events: generating a payment QR, setting up a Vipps collection, and exporting data to their accounting. It will show step-by-step how to link the event sales or donations to bookkeeping (e.g. "Scan Vipps QR at the door, then import the settlement report into your ledger software"). The guide will reference our modules (without requiring them) and demonstrate the benefits of automation. By helping clubs organize better, we build goodwill and may attract them to our automated solutions later.

## Conclusion and Recommendations

Vipps MobilePay provides all the necessary APIs to streamline payments and bookkeeping, but realizing these integrations is a technical task. All communication with Vipps is done over secure APIs, and accounting systems have their own REST interfaces. Combining them requires careful handling of authentication, data formats, and business logic. As Vipps notes, "all APIs use the same API keys" and integrating multiple flows is supported [1] [2], which simplifies development. However, it still demands an experienced developer to implement error handling, daily batching, and reconciliation logic correctly. Therefore, we recommend that Averdi's firm hire or assign a **dedicated developer** for this project. This developer will implement the modules above, test all payment flows, and set up the automated data sync.

By building these Vipps integration modules, Averdi can offer accounting clients a value-added service: **no more manual entry of Vipps payments**. Sales and donations paid via Vipps will feed directly into the books, saving time and reducing errors. The subscription-based model ensures ongoing support. In summary, our solution leverages Vipps MobilePay's comprehensive APIs [3] [4] and the accounting systems' capabilities [11] [10] to make bookkeeping "veldig, veldig enkelt" – very, very easy – for Averdi's clients.

**Sources:** Vipps MobilePay Developer Documentation [3] [5] [4] [7] [9] [1] [2]; PowerOffice API docs [11] [14]; 24SevenOffice integration page [10].

[1] [2] API platform | Vipps MobilePay Developer Docs
https://developer.vippsmobilepay.com/docs/APIs/

[3] [6] Introduction to the Checkout API | Vipps MobilePay Developer Docs
https://developer.vippsmobilepay.com/docs/APIs/checkout-api/

[4] [8] Introduction to the Recurring API | Vipps MobilePay Developer Docs
https://developer.vippsmobilepay.com/docs/APIs/recurring-api/

[5] Introduction to the Donations API | Vipps MobilePay Developer Docs
https://developer.vippsmobilepay.com/docs/APIs/donations-api/

[7] Introduction to the ePayment API | Vipps MobilePay Developer Docs
https://developer.vippsmobilepay.com/docs/APIs/epayment-api/

[9] Introduction to the Report API | Vipps MobilePay Developer Docs
https://developer.vippsmobilepay.com/docs/APIs/report-api/

[10] Integrasjon mellom 24SevenOffice og Vipps | 24SevenOffice
https://24sevenoffice.com/no/integrasjoner/vipps

[11] [12] [13] [14] [15] Workflow/Usecase/ecommerce_pos_payment/payment-events - PowerOffice API
https://developer.poweroffice.net/workflow/usecase/ecommerce_pos_payment/payment-events