# Integration & Migration guide from AfP Managed to AfP Partner Model

This is an overview of the steps that you need to follow to migrate your existing customer base from AfP Managed to the Balance Platform.

This guide focuses on addressing the steps specific to the AfP Managed to Balance Platform migration.

This process is split in three phases, a preparation phase followed by the actual migration and a final cleanup step if needed.

# Overview

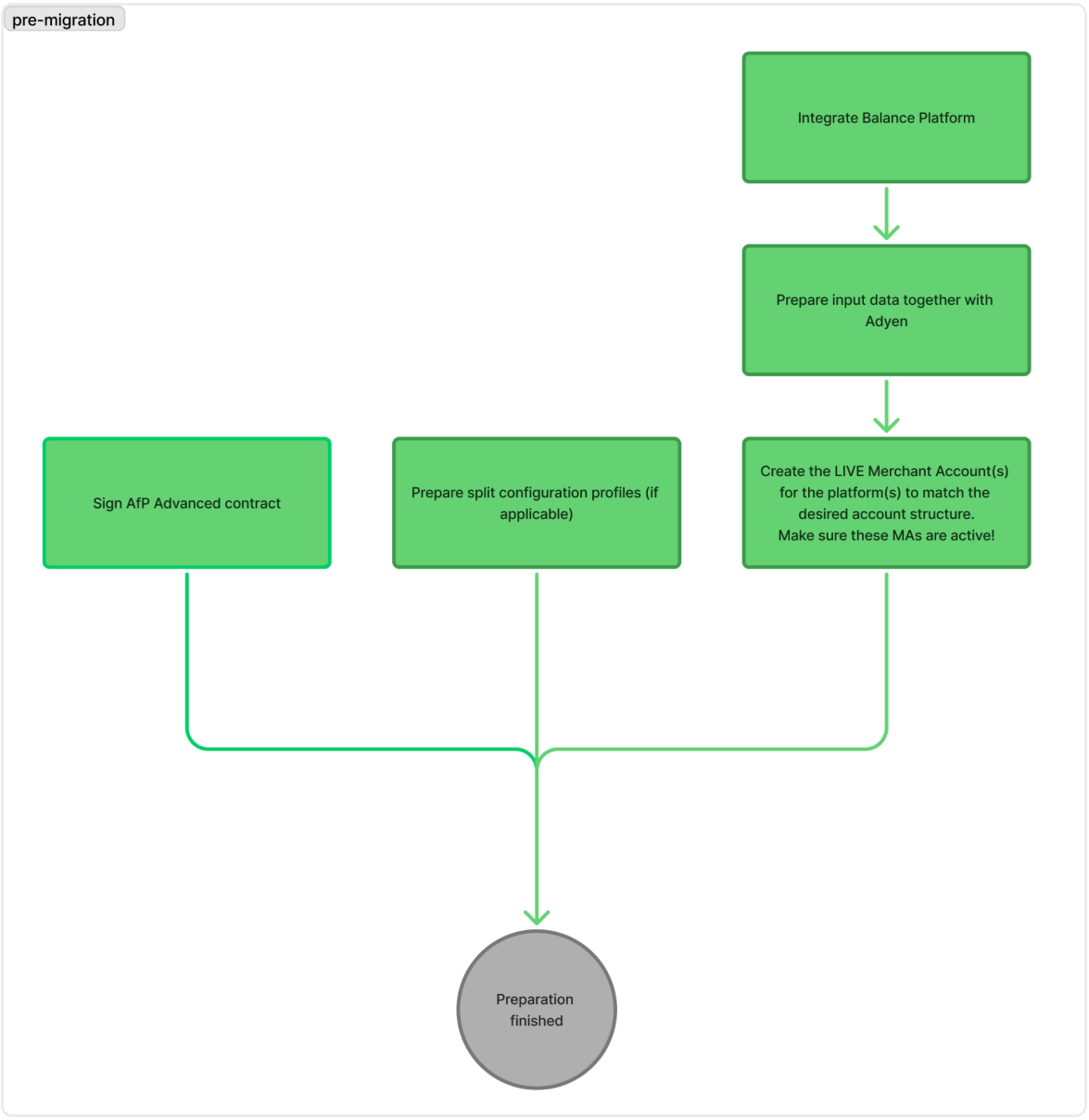
## What has to be done on a high level?

The main idea is for the migration to be completed in **two main phases**:

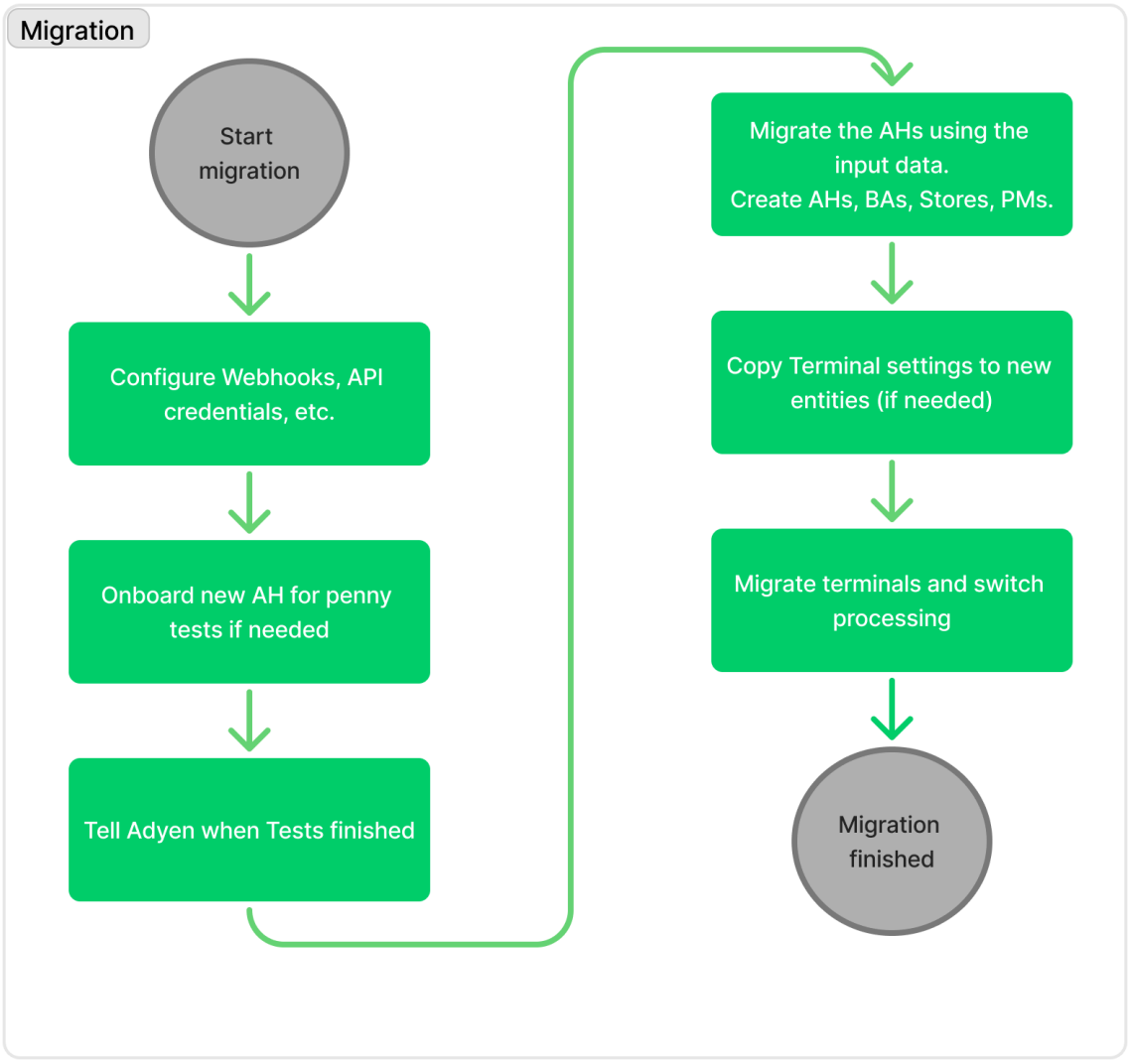
1. **Preparation Phase:** Submerchants will continue their current processing methods while we prepare all necessary components to ensure a seamless transition.
2. **Migration Phase:** The customer data is fully migrated, aiming for the smoothest possible handover based on the prior preparation."

Here are some flow charts to give an overview of the steps required.

## Integration and migration preparation:



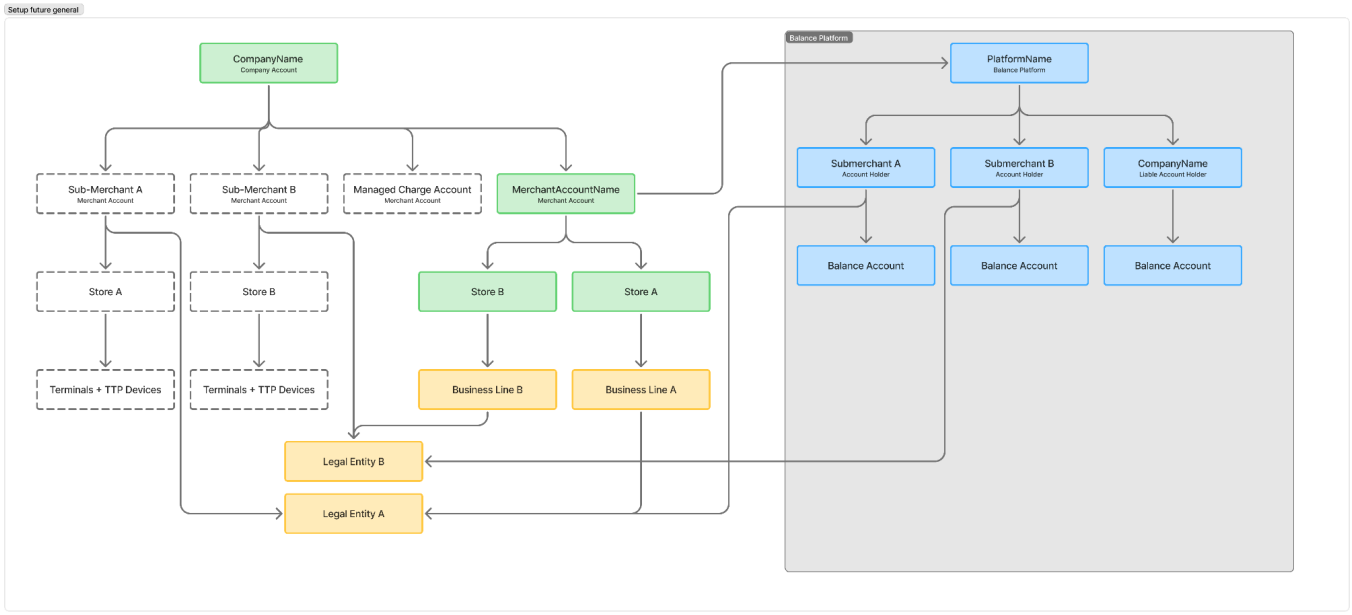
**Migration**




## Current and future structure

**Current Setup:** On AfP Managed every submerchant is represented using a Merchant Account.

**Future Setup:** When using a Balance platform there is just one (or a small set of) Merchant Account(s) needed. Instead of a Merchant Account every submerchant gets one or more stores depending on their business needs. These stores are then linked to the existing and already KYCd legal entity through a Business Line.




On the Balance Platform every submerchant and the platform operator gets an Account Holder and one or more Balance Accounts which hold the funds.

In the course of migration the existing terminals and Tap to Pay (TTP) devices have to migrate from the existing merchant accounts and stores to the new structure.

# Pre-migration phase

## Platform Integration

To be ready to migrate existing business over to a Balance Platform you must be able to properly operate a Balance Platform. For this your assigned Implementation Engineer will guide you through the integration.

In our docs we have a section covering your platform model and explaining all required tasks. Since you are already transacting with Adyen several steps can be skipped or just require some changes.

Get familiar with the new account structure and the new entities you need to deal with. There is also a webinar explaining the structure in detail. Do not continue until you understand this structure because it is crucial for the following chapters.

The following is just a general and high-level recommendation, speak to your assigned Implementation Engineer to find the best setup for your business.

**Onboarding**
We recommend you using our Hosted Onboarding solution. This allows you to collect all required documents and data points for all your sub-merchants without the need to build the forms yourself.

On this page, Adyen takes care of the onboarding process and the user interface (UI). Adyen offers two different ways to implement hosted onboarding:

- Onboarding on invite: In your Customer Area, request Adyen to create an account holder for your user. You then create a hosted onboarding link to redirect your user to enter their information.
- API-initiated: Create resources and a hosted onboarding link by making API calls to several endpoints, and redirect your user to enter their information.

**Payments**

Currently a pricing plan is configured for each of your sub-merchants which defines the “sell-rate” you sell payments for. Your customer then pays this sell rate and the difference between your “buy-rate” and the rate your customer pays is your commission.

On a balance platform this works differently. For every transaction you process, you can define instructions on how to split the funds between the balance accounts in your platform. This way, you can ensure that the sale amount, your platform's commission, the transaction fees, and other amounts are booked to the right balance accounts.

You can either define these instructions manually for each online or in-person request you send, or you can set up split configuration profiles that automatically split all transactions processed through a specific store. The instructions you provide in your API requests override the instructions you defined in your split configuration profiles.

We recommend you use split configuration profiles which can be seen as a replacement to the pricing plan on AfP Managed - you configure them once for your customer and they are applied to all transactions automatically. Speak to your Account Manager in case you need help translating your pricing plans into split configuration profiles.

ECOM Payments: You need to process over the new `merchantAccount`(s) and submit a `store` for each payment.

POS transaction requests do not change at all.

**Payouts**

When a shopper completes a purchase with one of your platform's users, the shopper's funds are first settled to your user's balance account. This means that Adyen received the funds and credited these to your user’s account. Your settlement schedule is the combination of the day on which the transaction takes place and the settlement delay.

Providing a reliable payout experience is critical to support your platform's users. You must set clear expectations about payout times.

For scheduled payouts, the payout times are automated by configuring sweeps that push out or pull in funds from a balance account based on a pre-defined schedule, amount, and source or destination. On-demand payouts let you initiate payouts off-schedule.

We recommend using scheduled payouts because they replicate what you currently have.

**Reporting and Reconciliation**

Reporting and reconciliation works differently compared to your current setup. The settlement details report now shows the full transaction amounts and not the part which is for the submerchant only. Therefore you need to change the way of reporting towards your customers. The Statements generation is also not possible any more which means that it has to be redefined during the integration.

Follow our docs for common flows used to reconcile payments and fees on a Balance Platform. Your Account Manager together with the Implementation Engineer can support you defining a reporting and reconciliation strategy.

There is also no Adyen hosted Dashboard your submerchant can log in to and retrieve their payment related data directly. Instead you have to build such a mechanism if you want to offer it. To make this integration easier we support with prebuilt components you can natively embed into your offering.

You can expect this to be the most complex and time consuming part of your integration.

## Change onboarding version

In order to be able to create a Balance Platform your Account has to be put on the latest KYC program. While this does not directly affect your submerchants there are technical implications you need to be aware of and able to handle.

While you can still use your AfP Managed integration as it is, we might be sending out new information via webhook. To have a smooth update of the KYC program version double check if your systems can properly handle the new webhook content. The easiest way would be to verify this using your TEST environment.

Speak to your assigned Implementation Engineer to come up with a strategy to bring your Account to the current version.

## Prepare input data for Migration

To be able to start the migration you need to make the mapping of the old entities to the new ones which will be created.

We will provide you the Legal Entity IDs of your current customers with the corresponding Merchant Accounts.

To prepare the migration you can leverage the following endpoints to retrieve the existing data from our system.

- **Retrieve the legal entity object:** Use `GET https://kyc-{env}.adyen.com/lem/v4/legalEntities/{LE-ID}`.
- **Get the business lines:** Use `GET https://kyc-{env}.adyen.com/lem/v4/legalEntities/{LE-ID}/businessLines`.
- **Fetch the merchant account:** Use `GET https://management-{env}.adyen.com/v3/merchants/{merchantAccount}`.
- **Retrieve the merchant's stores:** Use `GET https://management-{env}.adyen.com/v3/merchants/{merchantId}/stores`.
- **Obtain the payment methods (and associated stores):** Use `GET https://management-{env}.adyen.com/v3/merchants/{merchantId}/paymentMethodSettings`
- **Obtain the terminals for a merchant account:** Use `GET`
    a. **All terminals:** `https://management-{env}.adyen.com/v3/terminals`
    b. **For one or more merchants accounts:** `https://management-{env}.adyen.com/v3/terminals?merchantIDs=TESTMerchant1234,TESTMerchant1234`
    c. **For one or more stores:** `https://management-{env}.adyen.com/v3/terminals?storeIds=TestStore1,TestStore2`

Through this you should be able to retrieve all required information. Additionally you can leverage the `POS terminal fleet Scheduled Report`.

## Make a migration plan

When making a migration plan think of the following topics. Your Implementation Engineer can help you shape an optimal plan to minimize the impact on your customer base.

Check the chapter about the actual migration of a submerchant below and prepare as much as you can to make the transition as smooth as possible.

#### Operational feasibility

When you have to migrate many terminals, submerchants might reach out to your support team in case things do not go smoothly. Therefore the team should be able to handle the expected

outreaches from the submerchants. Make sure to properly instruct your team(s) and prepare them with appropriate material.

**Timing**
Since reporting will be a mess in the month(s) of migration, aim to finish the migration within a calendar month. This ensures that the impact on reporting/reconciliation is as small as possible. While not always feasible, we recommend starting the migration at the beginning of a month.

**Batch definition**
You might want to do the migration in multiple batches. The submerchants which are migrated in a certain batch should be distributed in a good way to e.g. reduce chance of support overload.

## LIVE Merchant Account creation

We need the merchant account(s) the platform will be linked to prior to platform creation. You need to request these accounts using your own legal entity information. Discuss with your Implementation Engineer how many merchant accounts you need and request them accordingly.

# Migration Phase

Now that all preparations are finished you can start the migration itself. For this, a LIVE balance platform is created first, then you will be doing your configurations and tests before you finally migrate your customer base.

## Step 1: Balance Platform creation and configuration

Your Implementation Engineer will create the LIVE platform(s) for you. Once this is done you can set up all the required things like API credentials, Webhooks and Report automations.

Double check that your existing API credentials you may want to keep using have access to the newly created Merchant Accounts to ensure a smooth transition.

There are several new roles on the Balance platform. Give these roles to all necessary users.

**Prepare split configuration profiles on the new Merchant Accounts**
You need to set up split configuration profiles which will be applied instead of the pricing plans after migration. Your Account Manager can support you transforming the pricing plans into split configuration profiles.

## Step 2: Onboard a net new or migrate an existing submerchant

In order to perform End to end tests we recommend you onboard a net new Account Holder to the newly created Balance Platform (you could simply onboard your own company). Additionally you can also migrate a first customer of yours to test the migration approach. If you want to migrate a submerchant follow the steps below.

Important: **Do not migrate more or onboard additional new accounts yet**, as we first need to do some configurations! Keep onboarding new accounts on the old AfP Managed platform.

## Step 3: Test test test

Make sure you test all scenarios you want to test. Also think about the unhappy flows like failing KYC checks, failing payments, etc.

## Step 4: Stop onboarding on the old platform

The next step is stopping onboarding of new customers on the old platform. Align with your Implementation Engineer beforehand to keep this timeframe as short as possible.

Once you fully stopped onboarding let your Implementation Engineer know who then will do the necessary configuration in the backend. This usually can be done within a couple of hours if properly planned ahead.

Once you get the confirmation that the configuration is done continue onboarding on the NEW Balance Platform!

## Step 5: Migrate all your submerchants

**Step 5.1: Create Account Holder reusing the existing Legal Entity**
First an account holder has to be created which is linked to the existing legal entity of the submerchant. This step will automatically request missing capabilities (like `sendToBalanceAccount` and `receiveFromBalanceAccount`).

**Step 5.2: Create Balance Account**
Create a Balance Account for the Account Holder created above as you would for any net new submerchant.

**Step 5.3: Create Store(s) reusing the existing Business Lines and apply a split configuration profile**
To retrieve the existing business lines of a merchant account leverage the https://docs.adyen.com/api-explorer/legalentity/4/get/legalEntities/(id)/businessLines endpoint. This should already be prepared in the pre-migration phase.

Make sure to not just create the POS stores but also the new ECOM ones if needed! If there is no Business Line for the industry/channel combination, create a new one.

**Step 5.4: Request Payment Methods**
Now request the payment methods which were also active on the corresponding Managed Merchant Account (a different set of methods is also possible of course).

Reuse the existing Business Lines here as well. If there are multiple for the same industry use the one which has most (or the best set of) channels.

Make sure to link the Payment Methods to the stores created above.

You can use the GET /paymentMethodSettings endpoint to retrieve the existing payment methods under a store/merchant account. This also gives you the currently attached business line.

Note: We do not offer STAR as a payment method on the Balance Platform but we do on AfP Managed.

**Step 5.5: Migrate terminal settings (if applicable)**
Copy the terminal settings of the existing Stores or Merchant accounts to the platform's store you created in the previous step. This ensures that all the settings and behaviours of the terminals stay the same.

**Step 5.6: Migrate Terminals and payments**
To migrate the terminals of the sub-merchant follow the steps below. Now you can also switch ECOM processing to the new MIDs. Make sure the API user is allowed to use the new Payment Methods.

This migration step can also be skipped here and everything can be mass migrated as described in the following chapter.

## Step 6: Migrate Terminals and Tap to Pay devices

**Step 6.1: Terminals**
Reassign all existing terminals to the corresponding stores.

**Note:** It is important that the API credential used to reassign the terminals has access to the old AND new merchant accounts! For simplicity we recommend using a credential on Company level which has access to all merchant accounts under this Company Account!

Depending on your integration different settings or steps might be necessary:

**Local vs cloud based connection**

Local connection

**Make sure that the encryption keys are set up correctly!**

You need to make sure that encryption can be successful. For simplicity it might make sense to

a) Configure the same encryption key platform wide on e.g. company level (maybe they do not want this for security reasons)
b) Configure the key on terminal level - through this it stays consistent when the underlying store changes

If you need to set these keys on store level you can do this of course, just make sure you use the same key as in you cash register system.

### Cloud based connection

We recommend using one API credential which can see both merchant accounts of a terminal (the new one linked to the BP and the old one). Through this you make sure that payments can be done successfully no matter the terminal is boarded to one or the other merchant account. For simplicity it might make sense to use one credential which is living on company account level and seeing everything.

#### Network connection

To retrieve the setup the terminals are using, use the Customer Area, our APIs or the `POS terminal fleet Scheduled` report.

**Important Note:** Offline payments will be lost if not synchronized before reassignment.

Reassigning terminals should preserve local network settings like Wi-Fi Configuration and static IPs stored on the device. However, conflicts with "remote" settings such as WiFi Profiles or hardware specific behavior can still cause disconnections.

### 1. Wi-Fi Connectivity

**Behavior:** Terminals generally retain manually entered Wi-Fi configurations during reassignment.

**WiFi Profiles vs. Manually Entered:**

- **Avoid Conflicts:** Ensure that both merchant accounts do not have a **Wi-Fi Profile** set.
- **Why?** A Wi-Fi profile overwrites manually entered configurations. The device will lose this profile, but then lacks the connection required to download the new profile.

**Hardware Specifics:**

- **High Risk:** Older devices (specifically older **Verifone V400cPlus** models) have shown inconsistent behavior, sometimes requiring the Wi-Fi password to be re-entered even if settings should have persisted. Note this might influence your Batch definitions in the pre-migration phase.
- **Low Risk:** Android-based devices and terminals with active SIM cards rarely encounter these issues.

**Operational Advice:** Staff should be briefed that while the process is designed to be seamless, they may need to re-select the Wi-Fi network and enter the password if the device cannot automatically reconnect.

### 2. Ethernet Connectivity

**Behavior:** Both **DHCP** and **Static IP** configurations stored on the device are preserved during the reassignment process. **Action:** Ideally, no user interaction is required. The terminal will reboot/update and resume connection using the same IP settings.

**Verify & Test:** Just as with Wi-Fi, there is a small risk of configurations being wiped. The individual terminal and IP configuration setup should be tested in advance to minimize connection issues during the wider rollout.

#### Terminal Boarding Strategy

Consider which boarding model fits your migration best:

- **Auto Boarding:** Requires no user interaction on the device screen. Best for mass migrations.
- **OTP (One Time Password) Boarding:** Requires an operator to enter a code on the device. Best if you need to strictly verify that specific terminals were assigned to the correct merchant account.

Your Implementation Engineer can discuss this with you further, if needed.

### Can I do refunds across Merchant Accounts?

**Answer:** The terminal will not preserve the transaction history and you can't trigger it manually on the device. If the terminal receives a refund request for the "old" merchant account via Terminal API (TAPI) it can successfully process it.

**Scenario**

- Day 1: A payment is made on *Terminal_A* (linked to *Managed_MA*).
- Day 2: *Terminal_A* is migrated (now linked to *BalancePlatform_Store_MA*).
- Day 3: The customer wants a refund for the Day 1 payment.
    - You cannot use *Terminal_A* to trigger this as a referenced refund from the device.
    - A refund request **via TAPI** coming from a POS system should be successful
    - you can also trigger refunds via the **Customer Area** or through an **API call** and reference the old *Managed_MA* where the original payment lives if needed.

### How can we check if all terminals were reassigned?

**Answer:** We have three methods for verification.

- **1. Customer Area:**
    - Navigate to **In Person Payments > Terminals**.
    - Use the **Store** filter to verify if any managed stores or merchant accounts still have terminals assigned.
    - Repeat for inventory& new stores to verify if Terminals were successfully reassigned

- **2. Via Report**
    - Use the `POS Terminal Fleet Scheduled Report` to check the current location of the terminals
- **3. Via API:**
    - Use the `GET /terminals` endpoint.
    - Filter the query by your new `merchantIds` AND/OR `storeIds` (e.g., `GET /terminals?merchantIds=Managed_MA`).
    - The response will list all terminals currently assigned to that store.*Optional: write a script to loop through your list of new stores and compare the results against your master list of terminals to confirm none were missed.*

#### Step 6.2: Tap to Pay

To migrate your Tap to Pay (TTP) devices follow these steps.

### Required Changes

**Backend Configuration Update:** You must modify the logic in your backend service that handles the Adyen `/sessions` request.

- **Action:** Update the payload to inject the **New** `merchantAccount` and `store` identifiers instead of the old ones.
- **Result:** When the TTP SDK receives the response, it extracts the new `sdkData` (containing a new `installationID` and `sessionID`), effectively binding the device to the new merchant account.

---

### Approach

**Active Migration: Forced Client-Pull or Server-Push** Force the app to invalidate the current session immediately via a signal from the backend.

- **Mechanism:**
    - *Pull:* The app polls an endpoint to check for config updates.
    - *Push:* The backend sends a Push Notification or SSE (Server-Sent Event) to trigger a session teardown.
- **Blocker:** Both methods require specific logic (listeners/polling) to **already exist** within the mobile app.
- **Implication:** If this logic is not currently implemented, you would first need to build and release a new app version and wait for 100% adoption before you could execute the migration. This makes it unviable for an immediate migration.

# Post Migration Phase

After the migration is done you might want to do some clean-up steps. Additionally we highly recommend double checking whether indeed everything was correctly migrated.

## Check if all terminals were reassigned

We have three methods for verification.

- **1. Customer Area:**
  - Navigate to **In Person Payments > Terminals**.
  - Use the **Store** filter to verify if any managed stores or merchant accounts still have terminals assigned.
  - Repeat for inventory& new stores to verify if Terminals were successfully reassigned
- **2. Via Report**
  - Use the `POS Terminal Fleet Scheduled Report` to check the current location of the terminals
- **3. Via API:**
  - Use the `GET /terminals` endpoint.
  - Filter the query by your new `merchantIds` AND/OR `storeIds` (e.g., `GET /terminals?merchantIds=Managed_MA`).
  - The response will list all terminals currently assigned to that store.*Optional: write a script to loop through your list of new stores and compare the results against your master list of terminals to confirm none were missed.*

## Clean up old account structure

- Deactive users that do not need access to CA anymore
- Close old merchant accounts
- Deactivate / delete API credentials, webhooks
- Clean up roles and permissions of your users
-

...