

# Global Architecture of exchanges

manager.one <> Core Banking Client



# Global Architecture of exchanges manager.one <> Core Banking Client

When purchasing one or more modules from manager.one's white label solution, the bank must provide some information to allow exchanges between the solution and the existing core banking.

This need is detailed hereafter, firstly through the application architecture used and finally by the list of APIs necessary for the implementation of the manager.one solution.

---

## Application Architecture

The manager.one application architecture of the services is organized into two clusters: the applications cluster and the bank card cluster

The kubernetes applications cluster contains at least 3 nodes that host the following services:

- API
- Back Office
- manager.one websites
- Customer Console
- Registration funnel
- Entity website

The kubernetes bank cards cluster contains at least 3 nodes [3 instances] that host the following service:

- Management of bank card authorizations

The 2 clusters, applications and bank cards, are placed in different subnets to ensure maximum security.

<https://azure.microsoft.com/en-us/overview/security/>

- The subnet of the card authorizations service communicates via VPN with the partner processor.

- The applications subnet communicates via VPN with the Core Banking Client

## List of APIs needed

The application has to communicate with the bank's Core Banking via APIs in real time, or communicate via batch files with delay.

It is strongly recommended to work with APIs, mainly to calculate real time available balance on the account.

### I. Registration tunnel

#### Creating an account with IBAN generation

When a customer completes a registration through the funnel, it is necessary to be able to provide a bank account number.

This involves creating an account in Core Banking to reserve a final account number.

#### Third Party Declaration for a given account

It is necessary to be able to declare trusted third parties for an existing account.

It is also necessary to anticipate the removal of access to an account from a third party.

## II. APIs for the client console

### List of accounts of a third party

When a Third Party connects to the client console, it must be possible to list the bank accounts attached to it in Core Banking.

### Real-time Balance of an account

For a given account, one must be able to know the balance at a precise T time, to issue authorizations for bank card operations.

## III. List of account operations

### List of account operations

For a given account, we must be able to retrieve the list of all the operations that have been carried out and also those that are upcoming.

## IV. Transfers

### Beneficiary management

If the bank needs to store the data of future transfer recipients (national and international), it must be possible to create, modify or remove them.

### Transfer management

For a given account, it is necessary to be able to create, manage, and cancel transfers.

File exchanges must comply with the ISO 20022 standard [see [https://www.iso20022.org/payments\\_messages.page](https://www.iso20022.org/payments_messages.page)].

## **V. Direct Debit management**

It is necessary to have the ability to list the withdrawals that will appear on an account and be able to challenge them.

It is also necessary to be able to list and revoke the mandates attached to the account.

## **VI. List of account statements**

In case the bank generates the account statements for its customers, it is necessary to have the possibility to list them and download them via the client console so the third-party can have consult them.

In case manager.one generates them, it must be possible to clearly identify the date-ranges of statements to which the operations are attached.

## **VII. Account bank card management**

A third party must be able to list and administer the bank cards attached to his account.

## **VIII. Notification of new operations**

manager.one will provide a 'webhook' for core banking to push notifications to mobile customers.

The main case is the notification of new operations.

# API Documentation

The documentation of APIs is available on manager.one's website.

<https://docs.manager.one/#documentation-api>

With the following base URLs:

Test environment <https://api-stg.manager.one/>

Production environment <https://api.manager.one/>

Catalog v1.0.1 includes the following APIs:

- Getting started
- OAuth2.0
- Pagination
- Account
- Operation
- Statement
- Transfer
- Beneficiary
- Beneficiary List
- Document
- Periodic Transfer
- Error
- Change Log

**WEBSITE**

[www.manager.one](http://www.manager.one)

**EMAIL**

[contact@manager.one](mailto:contact@manager.one)

**OFFICE**

manager.one - Banque Wormser Frères  
13 Boulevard Haussmann  
75009 Paris

**PHONE**

01 47 70 83 83

Manager.one is a "Banque Wormser Frères" brand, limited company with a 16 million euros capital, RCS Paris B 562 102 913. Its head office is based in the 13 boulevard Haussmann in Paris 75009, authorized as a credit institution by the ACPR (4 place de Budapest, in Paris 75009) and registered as insurance or reinsurance broker by the ORIAS under number 07034699 and by SaGa CORP, simplified joint stock company with a 156688€ capital. Its head office is based in the 56 boulevard Flandrin in Paris 75116, registered under number 822 296 794 by the trade and company register of Paris and registered by Orias under the n°18008507 as exclusive agent in banking operations and payment services.