

Use cases

This chapter provides information about use cases available in the Crypto Transfers service. Crypto Transfers service enables a range of processes related to transferring funds using cryptocurrencies. In order to use the processes provided by the Crypto Transfers service, the Partner must have a **master balance** set up in the Verestro system.

Important! The service is still under development. Functionalities will be made available and added to the documentation on an ongoing basis.

Note: The blockchain networks supported by our partner Financial Institution are **Ethereum** and **Tron**.

Account management

This chapter provides information on **creating and managing crypto addresses**, as well as retrieving the information required to initiate transfers of funds originating from cryptocurrencies.

Note: Verestro itself does not support cryptocurrencies. Cryptocurrency operations are orchestrated by our partner **Financial Institution**, which is responsible for converting cryptocurrencies into FIAT currencies.

Our partner Financial Institution currently supports **three cryptocurrency symbols**, which are listed in the table below.

Crypto currency supported by Financial Institution (symbol)	Corresponding FIAT currency
USDT	USD
USDC	USD
EURC	EUR

Crypto address creation

This section describes the process of creating a **crypto deposit address**. To create a crypto deposit address referencing an end user's balance, the end user must be registered in the Verestro system and have a balance in the corresponding **FIAT currency**. For example, to create a crypto deposit address for **USDC** or **USDT**, the end user must have a balance in **USD**. A user with multiple balances will have a separate crypto deposit address for each balance, even when the same cryptocurrency symbol and blockchain network are used. The diagram below describes the crypto address creation process step by step.

Important! Please remember that if you want to initiate a top-up using a cryptocurrency such as USDT or USDC, the top-up must be sent to the crypto address associated with a USD balance. Otherwise, the transfer will **not** be successful. The same applies to EURC: it must be sent to the crypto address associated with an EUR balance.

Expand to see sequence diagram describing crypto address creation process step by step

```
@startuml
skinparam ParticipantPadding 30
skinparam BoxPadding 30
skinparam noteFontColor #FFFFFF
skinparam noteBackgroundColor #1C1E3F
skinparam noteBorderColor #1C1E3F
skinparam noteBorderThickness 1
skinparam sequence {
ArrowColor #1C1E3F
ArrowFontColor #1C1E3F
ActorBorderColor #1C1E3F
ActorBackgroundColor #FFFFFF
ActorFontStyle bold
ParticipantBorderColor #1C1E3F
ParticipantBackgroundColor #1C1E3F
ParticipantFontColor #FFFFFF
ParticipantFontStyle bold
LifeLineBackgroundColor #1C1E3F
LifeLineBorderColor #1C1E3F
}
actor "End user" as user
participant "Customer application" as issuer
participant "Verestro Crypton" as cry
participant "Verestro Antaca" as ant
note right of user: End user wants to create\nnew balance (KYC success)
user->issuer: Create new balance
issuer->ant: Create new balance (userId)
```

note left of ant: Create balance for user
issuer<-ant: Return created balance data
user<-issuer: Display balance
note right of user: End user wants\nto see all their balances
user->issuer: Show balances
issuer->ant: Get end user balances (userId)
ant->issuer: Return list of balances of end user
user<-issuer: Show end user balances
note right of user: End user chooses to top up\nselected balance with crypto
user->issuer: Top up my balance
issuer->cry: Get symbols for selected balance (balanceId)
cry->issuer: Return list of supported symbols (e.g. EURC)
issuer->user: Show the list of supported symbols
note right of user: End user selects in which symbol\nbalance should be top upped
user->issuer: Select symbol
issuer->cry: Get supported networks for the selected symbol (symbol)
issuer<-cry: Return supported networks (e.g. Ethereum)
user<-issuer: Show supported networks for the selected symbol
note right of user: End user chooses to create crypto address\nin selected network
user->issuer: Create new crypto address
issuer->cry: Create crypto address in the context\nof the selected network (balanceId, symbol,
network)
note left of cry: Create crypto address
cry->issuer: Return created crypto address
user<-issuer: Display crypto address
@enduml

Revision #24

Created 10 September 2026 09:55:03 by Jakub Kotyński

Updated 14 September 2026 07:55:44 by Jakub Kotyński