



Paymium Remittance protocol

API for transactional inter-exchange remittance

Type	Protocol RFC
Confidentiality	Confidential
Version	0.2-DRAFT
Date	07/07/2015
Authors	David FRANCOIS < david.francois@paymium.com > Romain BOUYÉ < romain.bouye@paymium.com >
Recipients	Public
Abstract	Description of the Paymium Remittance protocol in terms of required API endpoints

Table of Contents

[Table of Contents](#)

[Introduction](#)

[High-level overview](#)

[Instant currency transfer](#)

[API technical description](#)

[Versioning](#)

[Endpoint URLs](#)

[Authentication](#)

[API metadata](#)

[Response parameters](#)

[Quote request for a money transfer](#)

[Request parameters](#)

[Response parameters](#)

[Money transfer commit](#)

[Request parameters](#)

[Response parameters](#)

[Cancellation of a requested quote](#)

[Request parameters](#)

[Response parameters](#)

[Transaction history](#)

[Request parameters](#)

[Response parameters](#)

[Balances](#)

[Response parameters](#)

[Error codes](#)

[Error codes](#)

Introduction

This document aims at specifying the technical and operational requirements to implement the Bitcoin-based Paymium remittance protocol. It is intended to provide a basis for community and professional feedback.

These requirements consist of API endpoints enabling two entities to negotiate a currency transfer between their clients using Bitcoin.

The protocol works by allowing entities to negotiate a Bitcoin exchange rate to supported currencies.

The settlement part is not the focus of this document as it can be performed in very different ways, depending on the business relationship of the source and target exchanges, it could take the shape of:

- contract-specific settlement terms by maintaining outstanding balances on both sides,
- automated per-transfer Bitcoin settlement,
- payment channels for high-traffic remittance corridors,
- balance maintenance of the source with the destination,
- any other business arrangement.

High-level overview

Instant currency transfer

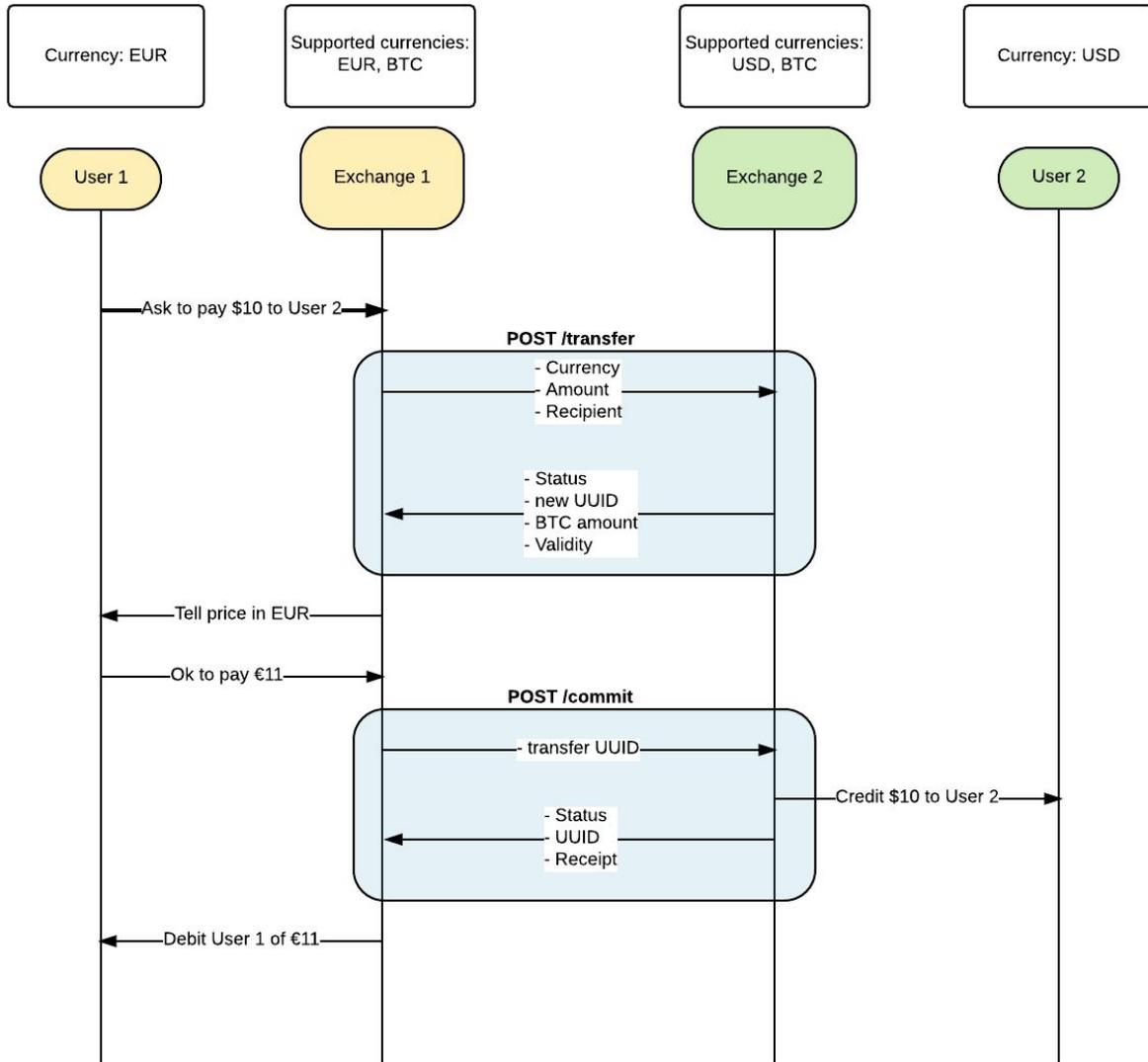
A user U1, registered on an exchange E1, wants to send a certain amount of money in a different currency to another user U2, registered on another exchange E2.

Given that:

- E1 only supports USD and BTC,
- E2 only supports EUR and BTC,
- U1 sends USD,
- U2 receives a fixed amount of EUR

Both exchanges must agree on an exchange rate setting a price for U1 in USD that ensures a given amount of EUR is ultimately delivered to U2.

Paymium Remittance protocol RFC



API technical description

Versioning

The base URL must include a protocol version string. It is not included in the specific endpoints URLs, but must be present nonetheless in the full URLs, and match the returned value by the “API metadata” endpoint.

Endpoint URLs

The URLs used for the endpoints must be named consistently according to the defined specification. Given a token and the base URL, an entity should be able to very easily connect to another peer.

Authentication

Authentication against the API should be performed using OAuth and HMAC'd tokens. Tokens are issued by one entity to another as part of the business relationship establishment.

API metadata

This endpoint shall be used to publicize the API capabilities and version.

Method	GET
URL	/info
Response example	<pre>{ "version": "v0.0.0-alpha", "currencies": ["EUR", "USD"], "options": ["CANCEL", "HISTORY", "BALANCE"], }</pre>

Response parameters

Name	Type	Req'd	Possible values	Description
version	String	Yes	v0.0.0	The API version
currencies	String	Yes	ISO currency codes	Which currencies the API supports

Quote request for a money transfer

This endpoint is used by the source exchange to get a quote from the destination exchange.

It is essentially a promise from the receiving entity that it shall credit the final beneficiary of the requested amount of currency if the sending party commits the transfer.

Method	POST
URL	/transfer
Request example	<pre>{ "dest_curr": "USD", "dest_amt": 128, "recipient": "john.doe@other-exchange.com" }</pre>
Response example	<pre>{ "status": "OK", "uuid": "15946fa0-24b8-11e5-9bf3-0002a5d5c51b", "btc_amt": 30000000, "valid_until": 1446280937 }</pre>

Request parameters

Name	Type	Req'd	Possible values	Description
dest_curr	String	Yes	ISO currency codes	The currency in which to credit the requested amount to the beneficiary.
dest_amt	Decimal	Yes	> 0	The amount to credit to the beneficiary
recipient	String	Yes	Defined by the recipient	The identifying string for the beneficiary

Response parameters

Name	Type	Req'd	Possible values	Description
status	String	Yes	"OK", error codes	The response status.
uuid	UUID	Yes		The transfer UUID
btc_amt	Integer	Yes	> 0	The amount in BTC satoshis that is required to pay in order to fulfill the original request
valid_until	Tstamp	Yes		The moment until which this particular quote will be honored if committed

Money transfer commit

Once a quote has been obtained, it can actually be committed in order for the final beneficiary to get credited by the destination exchange.

It's important to note that at this point, a debt between the exchanges is created. Being a business matter, and not a technical one, the settlement of this debt is not in the scope of this document.

The destination exchange must manage its risk and exposure to the source exchange in such a way that the final beneficiary credit is final and non-reversible.

Method	POST
URL	/commit
Request example	<pre>{ "uuid": "15946fa0-24b8-11e5-9bf3-0002a5d5c51b" }</pre>
Response example	<pre>{ "status": "OK", "uuid": "15946fa0-24b8-11e5-9bf3-0002a5d5c51b", "receipt": "<human-readable signed receipt>" }</pre>

Request parameters

Name	Type	Req'd	Possible values	Description
uuid	UUID	Yes		The UUID of the transfer we wish to commit.

Response parameters

Name	Type	Req'd	Possible values	Description
status	String	Yes	"OK", error codes	The response status.
uuid	UUID	Yes		The transfer UUID
receipt	String	Yes		The signed receipt for this transfer

Cancellation of a requested quote

For various reasons, it may be useful for the source to be able to cancel a quote that has been issued and is still valid. For example if the destination exchange manages its risk by allowing only a certain aggregated amount of outstanding quotes for a given source exchange.

A money transfer may be cancelled by the sender before it has been committed. Further attempts at committing the transfer must return an "SRC_CANCEL" error code.

This API endpoint is optional, support is indicated by the "CANCEL" flag in the "options" field of the "API metadata" call.

Method	DELETE
URL	/cancel
Request example	<pre>{ "uuid": "15946fa0-24b8-11e5-9bf3-0002a5d5c51b" }</pre>
Response example	<pre>{ "status": "OK" }</pre>

Request parameters

Name	Type	Req'd	Possible values	Description
uuid	UUID	Yes		The UUID of the transfer we wish to cancel.

Response parameters

Name	Type	Req'd	Possible values	Description
status	String	Yes	"OK", error codes	The response status.

Transaction history

An entity can ask for the transactions history. Either for all or a specific user.

This API endpoint is optional, support is indicated by the “HISTORY” flag in the “options” field of the “API metadata” call.

Method	GET
URL	/history
Request example	<pre>{ "cutoff": "828000", "recipient": "john.doe@other-exchange.com", }</pre>
Response example	<pre>{ "transactions": [{ "uuid": "15946fa0-24b8-11e5-9bf3-0002a5d5c51b", "dest_curr": "USD", "dest_amt": 1.28, "recipient": "john.doe@other-exchange.com", "btc_amt": 47307692, "status": "OK", "committed_at": 1436366919, "receipt": "<...>" }] }</pre>

Request parameters

Name	Type	Req'd	Possible values	Description
duration	Integer	Yes		Amount of time in seconds from now to fetch history.
recipient	String	No		If unspecified, the response should be the history for all users.

Response parameters

Name	Type	Req'd	Possible values	Description
uuid	UUID	Yes		The UUID of the transfer.
dest_curr	String	Yes	“USD”	The currency in which the beneficiary has been credited.
dest_amt	Decimal	Yes	1.28	

Paymium Remittance protocol RFC

recipient	String	No		The identifying string for the beneficiary. No required if present in the request..
btc_amt	Integer	Yes	> 0	The amount in BTC that is required to pay in order to fulfill the original request.
status	String	Yes	"OK", error codes	The response status.
committed_at	Tstamp	Yes		The moment when the transfer has been committed. NULL if the transfer is not committed.
receipt	String	No		The signed receipt for this transfer. Not required if the transfer failed or was cancelled, no receipt has been emitted

Balances

This API endpoint informs the source about its outstanding debt or available balance, as accounted for by the destination.

This API endpoint is optional, support is indicated by the “BALANCES” flag in the “options” field of the “API metadata” call.

Method	GET
URL	/balances
Response example	<pre>{ "btc": 2000000000, "at": 1436518782 }</pre>

Response parameters

Name	Type	Req'd	Possible values	Description
btc	Integer	Yes		Current balance of the entity account at the timestamped date.
at	Tstamp	Yes		Date of the balance calculation

Error codes

The status of a response must contain a meaningful error code if it is not “OK”. The currently specified error codes are detailed below.

Error codes

Code	Description
NO_SUCH_ACCOUNT	The beneficiary account was not found
INSUFFICIENT_MARGIN	There is not enough currently deposited collateral to fulfill the request
EXPIRED	The transfer is expired
SRC_CANCEL	The transfer has been cancelled by the sending entity
DST_CANCEL	The transfer has been cancelled by the receiving entity
ALREADY_EXECD	The transfer has already been executed
NO_SUCH_TRANSFER	The transfer corresponding to the requested UUID was not found
RATE_LIMIT_REACHED	The API maximum calls as been reached for the current window
TRANSFER_LIMIT_REACHED	The used reached his transfer limit
UNKNOWN_ERROR	General error