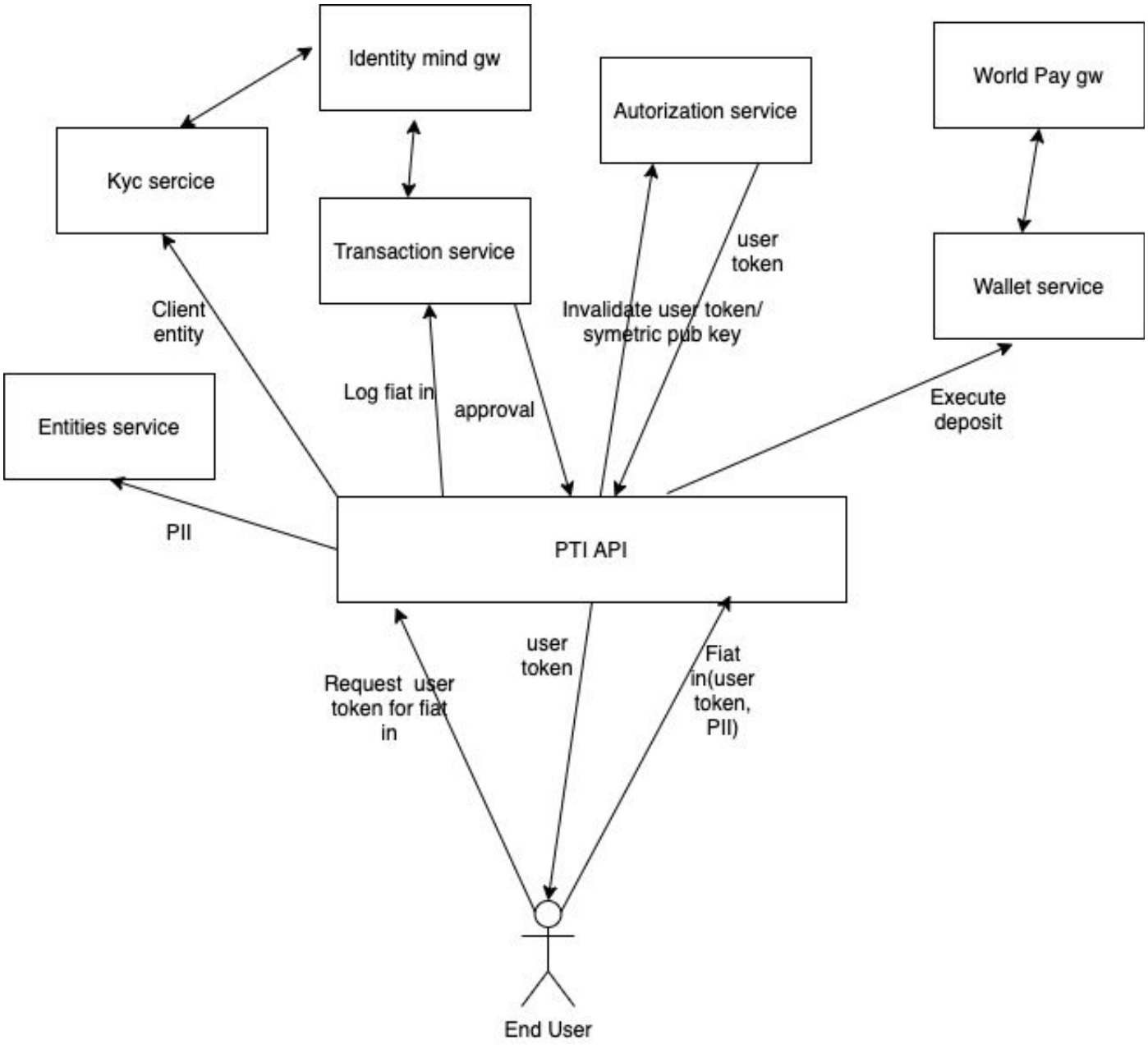
# Overview diagram fiat in




The end user in that example case would be the client user. Probably a client provided UI, that would call PTI backend directly using PTI provided “User token” concept.

## Prerequisites

To be able to request a user token, your user must be registered to PTI using the Public api. You must provide your own id for your user. The same goes for backend operations

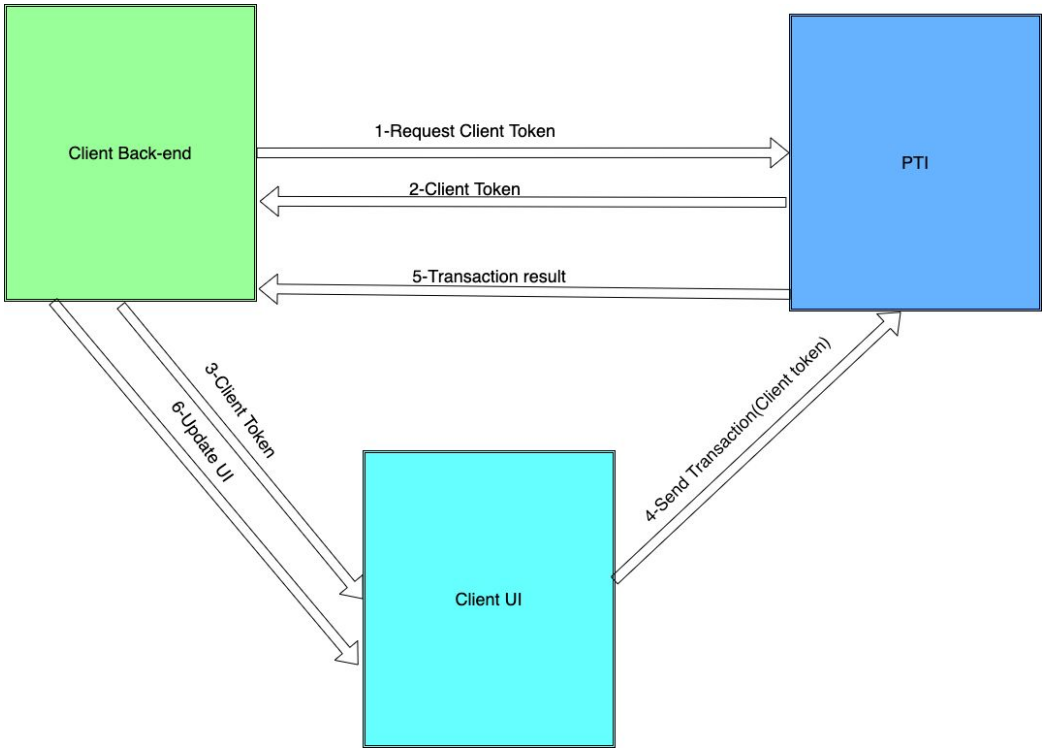
## User Authentication vs Client Authentication

User authentication results in getting a user token, which is a temporary access grant that exists only for the purpose of the end user of your application being able to perform one operation.

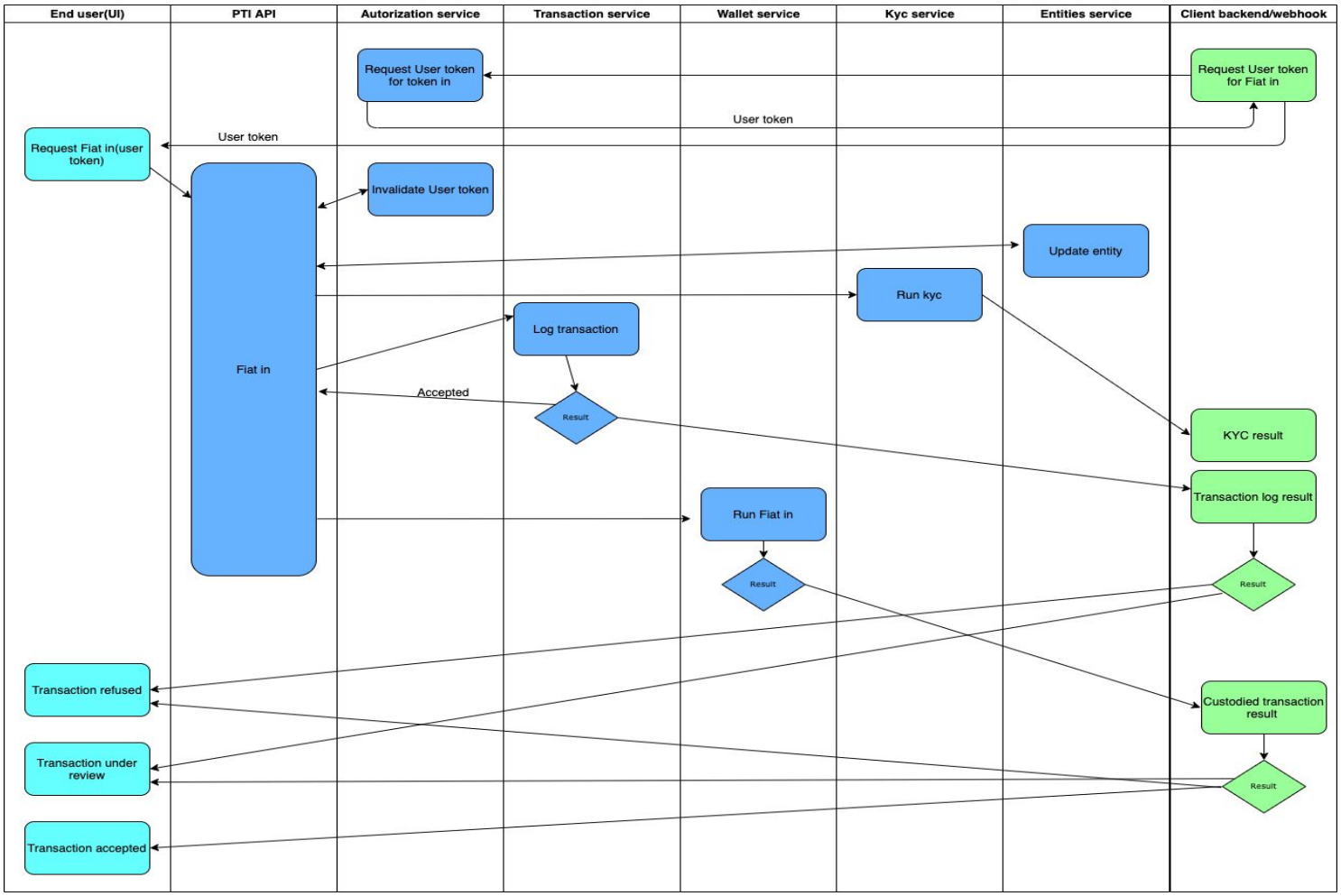
Client Authentication is performed using your private key, and gives you access to the PTI api for your backend. You need to sign each request to PTI using your private key. This is what gives you access to it, and therefore, constitutes the authentication process.

## How to use User token

You must use the following endpoint to obtain a user token: POST /auth/userToken
You must include the url you want to call as part of the body of your request.
Only operations availables will be issued a token. This operation is performed in the PTI client's backend.You must and include the user token in the request you want to perform on PTI backend.



# Detailed flow of a fiat in client side




## What happens when a client side Fiat in is initiated.

1. Client backend requests User token for Fiat in to PTI
2. PTI sends User token to client backend.
3. Client backend gives token to client front end.
4. End user fills the transaction forms, provided by the client
5. Client front end calls Fiat in API, with token, and receives an acknowledgment.
6. PTI validates token and invalidates it for one time use.
7. PTI takes the UI sent PII and updates the client entity with the new information if necessary.

8. PTI updated the KYC on that entity if changes have been made, for fiat in, this is just a background process, the KYC result does not affect the transaction
9. PTI logs the transaction and therefore, transaction monitoring routines are triggered(Id mind as of now), if the transaction is accepted, the process continues, if not, review or refusal are sent back to the caller.
10. PTI proceeds to execute the transaction in the wallet service. Payment gateway is called and if the transaction is accepted, the wallet is updated. Transaction results are sent back to the caller.
11. Asynchronously, kyc, transaction logging and custodied transaction results are published to the client's backend on its webhook. These results do not include PII.
12. Client backend updates the frontend with the result.

# Fiat out to third party case

PTI clients have the possibility to use PTI in order to register different third party users that ultimately relate to the same real world entity(It is PTI's job to reconcile them afterwards). That enables them to possess different fiat wallets(one for each third party user).

So in the case where a given PTI client wants to payout a third party with it's users wallets, the client is in possession of the related users, and so is PTI, and thus, can ask PTI to payout on them.