# Transaction overview

A platform that only allows clients submitting their current and expected consumption does not magically balances the grid. To do so

## Client-side trading information

Per every round of transactions all the clients submit their data to the blockchain using the app.DeliverTx(tx) (where the argument ‘tx’ is a bytearray representation of the data, an example is displayed below. At time data consists of the consumption and the production of the previous x minutes. Furthermore, it sends the prediction of the production and consumption for the following y minutes and the flexibility it has to offer for the next X minutes both consumption and production wise.

Since the price per offered energy unit may not be constant (A client may offer the first x kwh for $0,50, but can offer an additional y units only if the reimbursement will be higher), the client can send its price curve as a piecewise linear function as the pair ‘€/KWh’ : ‘Wh’. A negative price means the consumer is willing to decrease the corresponding consumption or production. A positive price indicates increasing the consumption or production.

‘MessageType’ : ClientReport,  
Data : { ‘UUID’ : 0010238408963,  
 ‘Timestamp’ : 123768238479,  
 ‘Consumption’ : 123.45 Wh,  
 ‘Production’ : 45.321 Wh,  
 ‘PredictedCons’ : {‘t+x’ : 232.43 Wh, ‘t+2x’ : 352.87, ‘t+3x’ : 32.54 Wh ... ‘t+y-x’ : 623.43 Wh},  
 ‘PredictedProd’ : {‘t+x’ : 4.433 Wh, ‘t+2x’ : 0, ‘t+3x’ : 32.54 Wh ... ‘t+y-x’ : 74.43 Wh},  
 ‘consFlexibility’ : {‘€-0,12’ : 110 Wh, ‘€-0,24’ : ‘ 55 Wh, ‘-0,85’’ : 20 Wh},  
 ‘prodFlexibility’ : {‘€-0,12’ : 12 Wh, ‘€-0,24’ : ‘ 55 Wh, ‘-0,85’’ : 20 Wh},

}

The offered energy in the example message above says that the client is willing to increase its consumption with 310wh in the period of x minutes for a payment of 12 cents per KWh. If grid is very imbalanced and requires even more, the client is willing to offer an additional 55 Wh for 24 cents per KWh and, if needed, another 20 Wh for 85 cent per Kwh.

## Trading platform side

After all the clients have submitted their data to the blockchain, and the information has been aggregated and propagated up the complete platform the matchmaking process can begin. Two matchmaking mechanisms are considered here, one is a drop-in replacement for the current situation and does not require flexible energy pricing by using the APX imbalance market. The second method is a more progressive mechanism that would completely disrupt the energy market as we know it.

### APX-based approach

The APX-based approach leaves the current energy market intact, but enables easier participating in the balancing. Here the TSO still sets the price and amount of energy to be regulated, however instead of first utilizing it’s buffers it tries to prevent the imbalance in the first place.

As the data was aggregated and propagated up the network, a complete picture of the current state of the grid was formed. The TSO can only view the clusters at the highest layer, but that’s all it needs since every cluster represents the expected production/consumption and flexibility of their sub-clusters. Since different clusters offered different prices for different capacities, the TSO may delegate the Up- and downregulation spread out across several clusters as it sees fit.

The cluster that got appointed a certain amount of energy to be balanced (along with the price point) by the TSO, receives the order along with the promise of a reimbursement. Every cluster will then spread out its appointed balancing and reimbursement across their sub-clusters and possible directly connected clients. Eventually all Balancing orders and their corresponding reimbursements will end up at a specific client who will then act according to its earlier submitted promises.

If there is any remaining imbalance the TSO may deploy the same methods as in the current situation. This approach is beneficial for both the TSO and the consumer, as the TSO can spend less on buy up- and downregulation capacity and the consumer can benefit by reducing his electricity bill (or even make money) by changing his behavior.

In this approach every client would still have a contract with a utility company that will supply the remaining energy. This mitigates the legal troubles of having variable energy pricing, as they are not charged for their consumption, but rewarded to adjust their consumption/production behavior. Also, participation of the trading platform is completely voluntarily, if a consumer does not have the capacity or has no interest in participating in this (possibly lucrative market) he can submit his flexibility to zero, for the single consumer this will result in being billed the same way he is in the current situation

## Truly distributed approach

The truly distributed approach creates a complete free market. Here all the data is aggregated and propagated in along the the same way as the APX-based approach. But for this approach to work there must be a single blockchain at the top of the chain where all the sub-clusters submit their data. Afterwards

As soon as all the data is submitted and propagated the highest blockchain will look at the total predicted consumption and production. Based on this knowledge using supply and demand mechanisms the electricity price for the coming time interval will be determined. This is the definite price that the customer will have to pay for the coming time interval. If there is surplus of energy the match making will look for anyone (clusters or clients depending on the level) who is willing to reduce the amount of consumption or increase their production based on the current price point and the offers made. These up- and downregulation or then spread out to sub-clusters (or clients) whom it concerns using the same mechanism as described above.

This would create a self-balancing market without a centralized body organizing it, but comes at the cost that the prices might fluctuate greatly. However, it will motivate clients to participate more heavily in balancing in times of extreme pricing, which are inherent to big imbalance.