|  |  |
| --- | --- |
| **Energy Imbalance Market & Blockchain**  CS4160 Blockchain Engineering (2017/18 Q2) | Project Report by:  Joseph Verburg 4018575  Michal Loin 4587324  Jetse Brouwer 4615964  Niels Hokke 4610148  Evgenia Domnenkova 4625633 |

# Introduction to Imbalance Market

The energy market consists of multiple markets with the main goal of balancing production and consumption. Each market works on a different timescale with different terms. The 4 most important markets are the Long-term market, day-ahead market, intra-day market and the unbalance market.

The imbalance market is the most volatile market. This market is led by the Transmission system operator (TSO, TenneT in the Netherlands) and serves the sole purpose of maintaining a stable frequency of the grid. When a imbalance arises TenneT has multiple reserves at his disposal. These reserves consist of parties who have the ability to regulate their consumption or production up or down relatively fast (depending on which reserve group they're in). The primary reserves will be used to solve immediate imbalance. The secondary reserve is the market that is used if the grid hasn't stabilized after the 15 minutes of using the primary reserves. At last there is a tertiary reserve, also called the emergency pool, for long term highpower regulation power.

# Problem Statement

With consumers of electricity now also becoming producers at a higher rate, more imbalance is created on the net. To maintain security of supply the imbalance should be kept within the limits of the grid. Currently the transmission system operator (TSO: Tennet in Netherlands) is responsible for maintaining this balance. A problem arises when the production becomes more and more decentralized, and the balancing is still the duty of a centralized body, it becomes harder to respond on the highly fluctuating consumption and production of the different actors.

The aim of this project is to design and prototype a decentralized market and a so called 'smart grid' where participants of the smart grid can dynamically match their supply and demand to minimize imbalances.

# General Approach to Balancing

The system should be able to be deployed in co-operation with the current electrical grid as it unrealistic to expect major changes in the grid topology in the forseeable future.  The grid is currently laid out in a manner where several households (1 or more streets) are clustered together sharing a physical cable. These clustered cables join at small brickhouse-substations spread out through the city. These substations are clustered together on a 10KV network, where one cluster roughly serves a city. Between nearby cities a 50KV net grid is clustering those together, from there they are clustered on a 110 -150 KV grid and again at a 380KV grid (which also connects with neighboring countries). In the current situation, the grid is balanced by the TSO, who buys capacity to up- or downregulate the consumption or production at companies spread out over several levels of the grid, and does so when required.

For balancing the grid matching consumption and production at the highest level is enough. Even when nationwide balance is achieved overload can happen due to unreasonably large imbalances on lower levels. Therefor we believe that balancing should, in the first place, happen within same cluster (e.g. a single street) before propagating it up to a higher level (e.g. district or city). When looking at an example of at the lowest, the spikey behavior of a single household is smoothened out by combining the households per cluster, and the smart grid aims to spread out the load equally over time. The remaining imbalance is then propagated to a higher level where bigger players (which more capacity but lower flexibility) can partake in the balancing game. For every step to a higher level the peaks become smoother and more predictable, which is very desirable for the big powerplants that will have to supply the energy in moments of energy shortage.

# Balancing Using Blockchain

To enable the actual balancing between the participants of the smart grid (which will be called clients from now on) a platform must be offered to gain knowledge of the current state, the expected state and to perform reach agreements on energy exchanges. For this project we will explore the possibilities of using a blockchain for providing such a platform.

As discussed earlier, the focus should first lie on balancing within a single cluster, therefor there is no need for connecting every client to every other client. This creates the possibility of creating a blockchain per cluster instead of having to design a blockchain for all households in the system, which reduces the amount of transactions per second per blockchain significantly.

The knowledge per cluster will then be aggregated and propagated as a single entry up to a higher level blockchain, where bigger parties can join the energy balancing game. The aggregation implies that individual information is lost when propagating to a higher level. However, this does not have to be a problem as the individuals in the cluster already did their best to solve the imbalance the best they could and the main task is now to balance out the remaining imbalance.

By following the physical lay-out for clustering the nodes we end up with a structure that could be visualized as a tree structure where the dataflows from node to node are only through their parent. A figure depicting such a tree is given in Figure 1. Here the circles containing the letter 'N' are the nodes that host the blockchain, the black symbols depict the clients (which can be households, industries and companies specialized in balancing).

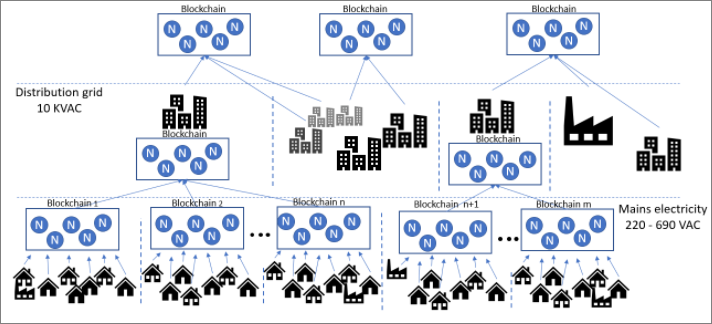


Figure 1: A possible data structure with a blockchain per cluster.

# Balancing Algorithm

Per every round of transactions all the clients submit their data to the blockchain. 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.

As soon as a blockchain has received a client report from all of its sub-clusters/clients it calculates its own cluster-specific price based on the imbalance within the cluster. Based on this price and the client’s flexibility the cluster can do a first balancing iteration. The remaining imbalance, along with the remaining flexibility, is then propagated one level up, the cluster on the higher level will again determine the price for that cluster based on the remaining imbalances and see how much it can balance based within its own cluster.

Using this mechanism the balancing starts with the focus on a single cluster, and will then spread out amongst its neighboring clusters every time the iterative process moves up a level. This would create a self-balancing market without a centralized body organizing it.

# Considerations

References and research?

Princeton block chain lectures: <https://piazza.com/princeton/spring2015/btctech/resources>

* [Blockchain market research](https://github.com/Tribler/tribler/issues/2559)
* [Blocklab paper](http://blocklab.nl/media/uploads/2017/09/Beta-Paper-Blockchain-x-Energy-online_opmaak.pdf?x54716)
* [Lecture notes Distributed Algorithms](https://github.com/NielsHokke/DEM/blob/master/LectureNotes-IN4150-20170109.pdf)
* [Master thesis of Sjors Hijgenaar](https://repository.tudelft.nl/islandora/object/uuid%3Ac4a8d2be-6c4e-4bfb-8a37-8a65bd7fe50b?collection=education)
* [Satoshi paper](https://bitcoin.org/bitcoin.pdf)

[Tendermint and PoS problems!](https://blog.cosmos.network/consensus-compare-casper-vs-tendermint-6df154ad56ae)

[Tendermint](https://tendermint.readthedocs.io/en/master/)

[Graph illustrating the consensus mechanism of Tenderminder](https://github.com/devcorn/hackatom/blob/master/tminfo.pdf)

[Tendermint thesis paper](https://atrium.lib.uoguelph.ca/xmlui/bitstream/handle/10214/9769/Buchman_Ethan_201606_MAsc.pdf)

[Blockchain as a service system based on Tendermint](https://github.com/monax/monax)

[Cosmos.Network](https://cosmos.network/faq)

[Etherium](http://www.ethdocs.org/en/latest/) [Casper](https://blockonomi.com/ethereum-casper/)

The team also looked into [Hyperledger-fabric](https://hyperledger-fabric.readthedocs.io/en/release/), [Ripple](https://ripple.com/), [Corda](https://www.corda.net/), [Iota](https://dev.iota.org/), [Waves](https://github.com/wavesplatform/Waves/wiki/Waves-Node-REST-API), [Bitshare](http://docs.bitshares.org/)

# Solution Overview

**Layers**

* Client Layer (i.e. Smart Meters)
* Blockchain Layer
* Visualisation Layer (Thin Client)

**Client Layer**

In Scope:

* Ability to generate transaction of specific format
* Ability to transmit the transaction to the Blockchain Layer
* Ability to combine smart meter reading, prediction agent output and trading agent output into one transaction as part of the transaction format
* Ability to sign transactions

Out of Scope:

* Connection to real-life smart meters
* Intelligent generation of prediction agent output and trading agent output

**Blockchain Layer**

Blockchain Method and Platform:

* Ability to run permissioned blockchain:
  + Ability to maintain similarity with the current physical infrastructure
  + Ability to hold consensus servers accountable
  + Ability to trace clients/users in the system
* Ability to run blockchain with BFT-based PoS design:
  + Ability to reach the performance level required by the target field of implementation with limited number of nodes (e.g. to serve up to thousands of clients with only up to 20 nodes)
  + In addition, BFT blockchains are per definition permissioned, requiring the sharing of identities amongst the servers, in support of the requirement related to permissioned blockchain
* Ability to be flexible over the blockchain application functionality so that any block/contract logic can be defined depending on the field of implementation

Blockchain integration with the Client Layer:

* Ability to receive transaction of specific format from the Client Layer
* Ability to add block to the blockchain dedicated to the Cluster L0

Blockchain functionality for Visualisation:

* Ability to provide data to the visualisation layer

Blockchain functionality for Cluster L0 Blockchain:

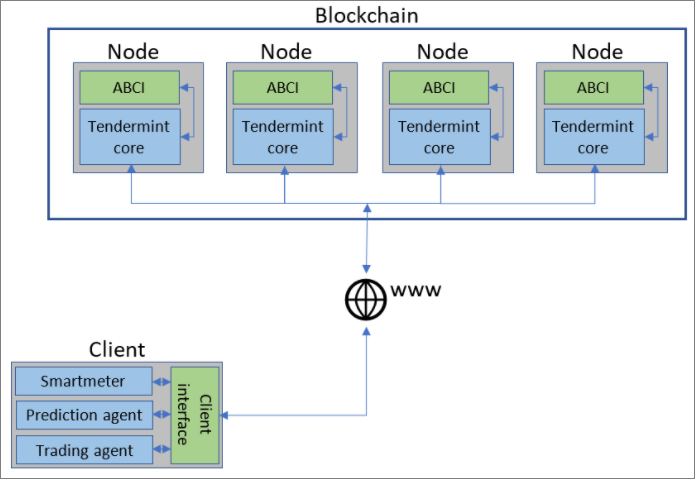
* Ability to maintain the logic of the blockchain – logic TBC

Blockchain functionality for Cluster L1:

* Ability to aggregate information for Cluster L0 Blockchain
* Ability to transmit transaction to the Cluster L1 blockchain

Out of scope:

* Security
* Smart Contracts



# Decisions

**Why use blockchain for grid balancing:**

* Balancing was selected as the use case as it fits the parameters of general blockchain use case:
  + Several parties without any one authority
  + Lack of trust among those parties
* Effective balancing of the grid is also one of the major challenges of the distributed energy market in a market where any consumer can also be a supplier

**Why Tendermint as the blockchain platform:**

* Ability to run permissioned blockchain for:
  + Ability to maintain similarity with the current physical infrastructure
  + Ability to hold consensus servers accountable
  + Ability to trace clients/users in the system
* Ability to run blockchain with BFT-based PoS design:
  + Ability to reach the performance level required by the target field of implementation with limited number of nodes (e.g. to serve up to thousands of clients with only up to 20 nodes)
  + In addition, BFT blockchains are per definition permissioned, requiring the sharing of identities amongst the servers, in support of the requirement related to permissioned blockchain
* Ability to be flexible over the blockchain application functionality so that any block/contract logic (ABCI application) can be defined depending on the field of implementation

Some technical challenges tough:

* Lack of technical knowledge and learnings from other developers available online due to Tendemint being a young technology
* Lack of Tendermint supporting material for Python developments
* Migration to new versions of Tendermint during the project
* General software development and integration challenges