



D5.2 Description of implemented toolset TSO

June 30, 2019

EcoGrid 2.0 is a research and demonstration project funded by EUDP (Energiteknologisk Udviklings- og Demonstrationsprogram). The 9 partners in the project are:



Main Authors:

Name/Partner	Email
Bernhard Jansen/IBM	bja@zurich.ibm.com
Peter Buhler/IBM	bup@zurich.ibm.com

TABLE OF CONTENTS

1 Introduction	7
2 Figure 1 - Component overview	Fejl! Bogmærke er ikke defineret.
3 Implementation description	8
<i>3.1 Implementation environment</i>	8
<i>3.2 Implementation</i>	9
<i>3.3 Runtime process</i>	10
4 Testing	12

1 Introduction

The TSO toolset and the implementation of it described in this deliverable is, together with the DSO toolset, an integral part of the EcoGrid 2.0 ecosystems containing the TSO toolset, the EcoGrid 2.0 Market, the aggregators and the participating houses as shown in **Fejl! Henvisningskilde ikke fundet..** The DSO toolset will be described in D5.1 and is not part of this deliverable.

The TSO toolset is a split effort between WP4 and WP5. WP4 developed the algorithm in *Task 4.2 Design of toolset for TSO* and documented it in *D4.2.1 Tool for Market Requests* and it is WP5 task to implement the algorithm in T5.2 and document it in *D5.2 Description of implemented toolset TSO*. This deliverable will not cover specifics of the algorithm, except where is necessary for the implementation description.

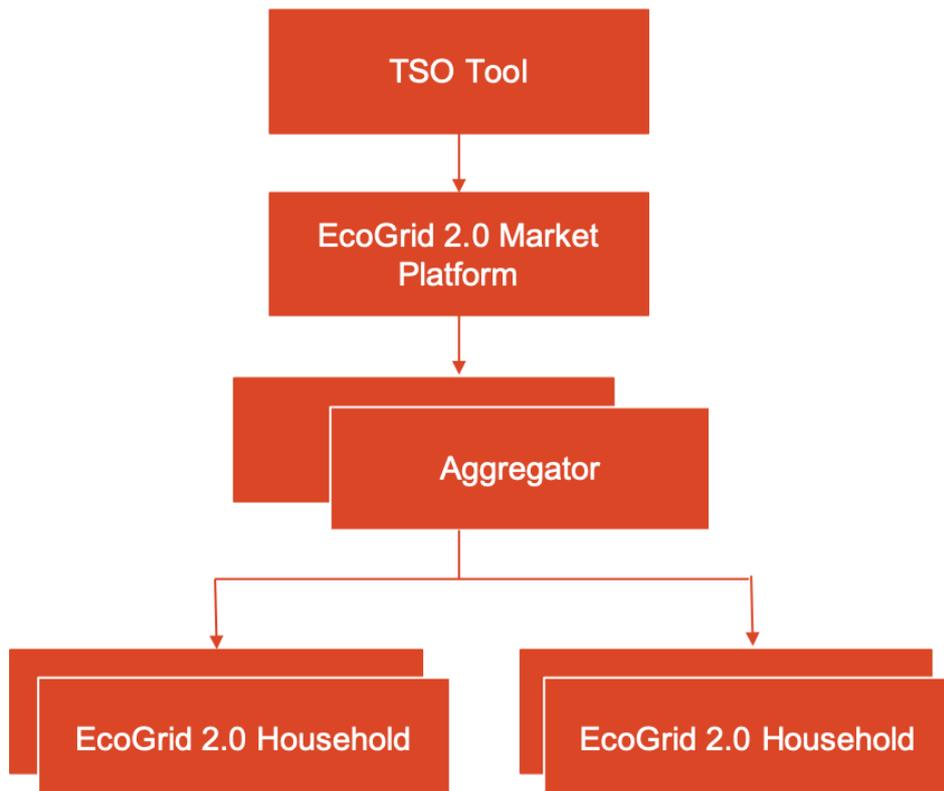


Figure 1 – Component overview

2 Implementation description

The original implementation of the TSO toolset was delivered as a MATLAB project containing two code files and a parameter file.

2.1 Implementation environment

The implementation of the TSO toolset as a usable part of the EcoGrid ecosystem was partly reusing the MATLAB code but mainly done in Java with the target platform IBM Cloud formally called IBM Bluemix. The IBM Cloud platform was chosen as the EcoGrid 2.0 Market is based on it and therefore the two implementations can run close to each other and the operation and maintenance is very similar.

The algorithm is comprised of two parts. The creation of the Markov matrices which serve as input to the imbalance forecasting and the imbalance forecasting itself. The creation of the Markov matrices is taken from the original code and extended to store the matrices in a format readable by the Java code used for imbalance forecasting. The matrices are created for a time span of first of June 2017 to end of May 2020 which covers the complete project period.

The imbalance forecasting part of the algorithm was translated from MATLAB to Java code to be executed every hour. The imbalance forecasting reads the pre-calculated Markov matrices and forecasts the imbalances for the next two hours. The calculated imbalance forecast is used to create and send a market opening request to the EcoGrid 2.0 Market; see D3.1.2 Market Interface Specification where the functionality and the formats of the market requests are described.

Figure 1 shows an overview of the development and runtime architecture.

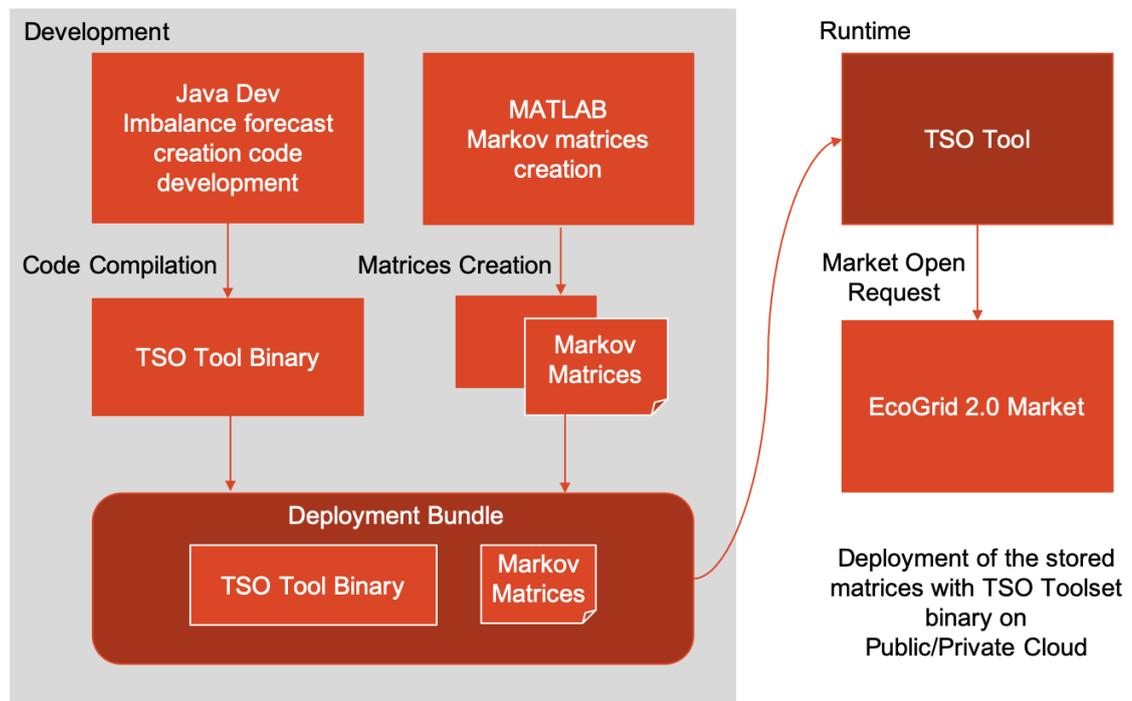


Figure 1 - Implementation and deployment overview

2.2 Implementation

Figure 2 shows a component diagram of the main components of the implemented TSO toolset. The components are following:

- **Scheduler:** The TSO toolset needs to run, as the EcoGrid 2.0 Market, 24x7 for the demonstration or operation period. The scheduler ensures that an imbalance forecast calculation is triggered at the beginning of every hour, timing is configurable, and the results of the imbalance forecast calculation are used to compile a new market open request which is then sent to the EcoGrid Market using the MQTT Client.
- **Imbalance Forecasting:** The imbalance forecasting module is the translated Matlab code from WP4 and calculates the imbalance forecast based on day, time of day and the input from the two Markov matrices.
- **MQTT Client:** The MQTT Client provides the communication channel to the EcoGrid 2.0 Market and handles the communication protocol overhead, the authentication and transport encoding and decoding.
- **Matrix Reader:** The two Markov matrices which were offline created in MATLAB and deployed to the runtime container in IBM Cloud need to be read at start-up and provided to the Imbalance Forecasting module. This module loads the matrices and provides it to the Imbalance Forecasting module.

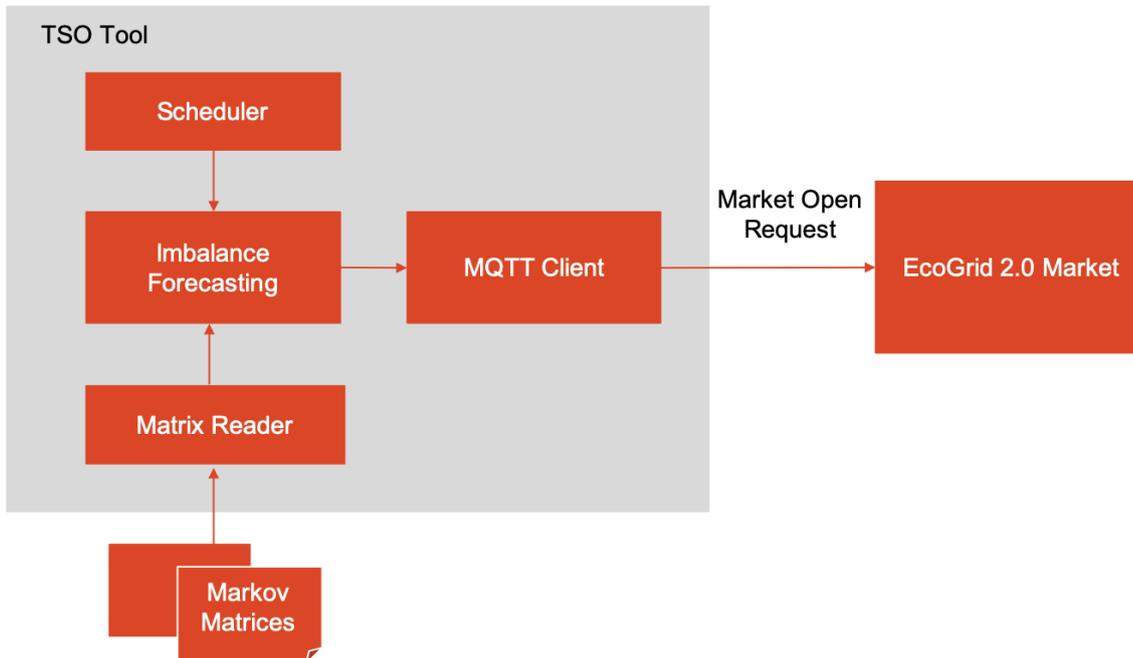


Figure 2 - Component diagram

2.3 Runtime process

As described above the TSO toolset needs to run on a regular basis, every hour in the current setting, to achieve this the implementation is running through the following process shown in

Figure 3.

On start of the TSO toolset the Markov matrices are loaded from the stored files into memory and provided to imbalance forecasting module. When loaded into memory the scheduler is started and the regular operation is started.

Once the scheduler triggers, the imbalance forecasting is started and calculates based on date, time of day and the input from the pre-calculated Markov matrices the imbalance for the next two hours. The imbalance is used to create a new Market Open Request which is sent subsequently to EcoGrid 2.0 Market; see D3.1.2 Market Interface Specification where the functionality and the formats of the market requests are described. If the TSO tool did not receive a stop request the process will start again with waiting for a trigger from the scheduler.

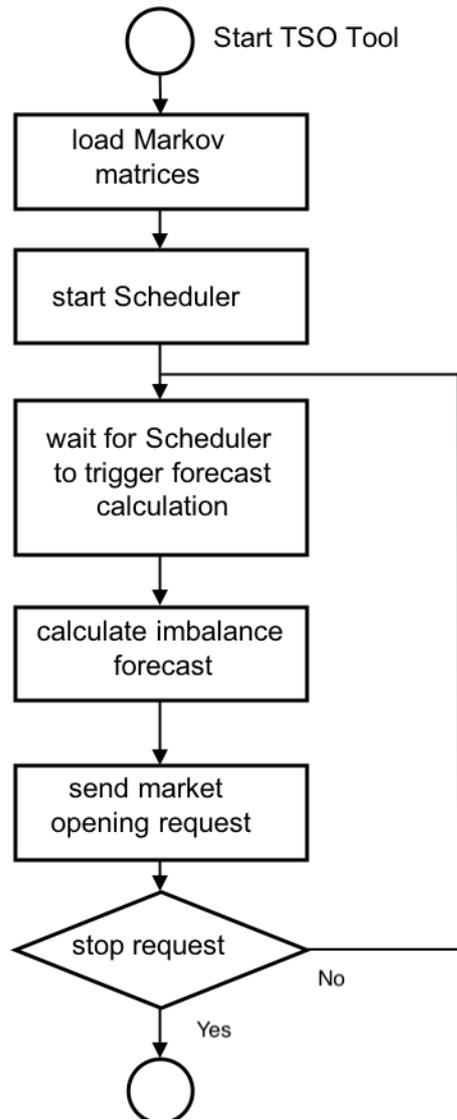


Figure 3 - Process flow

3 Testing

Tests were carried out to ensure the robust operation of the scheduler and the correct operation of the matrix loader as well as for the MQTT client. Verification of the results of the TSO tool were performed to ensure that they correctly reflect the imbalance forecasting produced by the algorithm provided as MATLAB code by DTU.

Read more at www.ecogrid.dk