An Open-Source Ethereum Smart Contract for Peer-to-Peer Decentralized Local Energy Market Transactions: DecEnergyM v0.0

Ali Arjomandi-Nezhad

Power systems Reconstruction alongside distributed generator technology advancements provides the consumers with the opportunity to become prosumers and make profits from selling their excessive power. However, with limited computation power of Distribution System Operator (DSO) and the huge number of Distributed Energy Resources (DERs) in the distribution system, central controlling the Local Energy Market (LEM), like what is performed at the transmission level, is almost impossible. In this regard, decentralized approaches are necessary for realizing LEM. Blockchain technology is one of the most promising platforms for this purpose. Several research works have proposed energy trading-specified blockchains for this aim. However, maintaining a secure blockchain is not a straightforward task. Besides, tokens of the energy trading-specified blockchains might be practically unusable. Therefore, in this open-source project, an Ethereum smart contract is developed for decentralized LEM. Not only is Ethereum a trusted and practically verified blockchain, but also its main token (Ether) is a valuable cryptocurrency that can be spent on various purposes. **Respected readers can find the full code of the smart contract in this link.** In the next paragraph, the structure of the proposed smart contract is elaborated on.

The schematic of the contract is summarized in Fig. 1. The DSO, which creates the smart contract, is specified as the owner of the contract in the constructor function. It can add addresses of accounts which are permitted to transact with each other. In case any transactor behaves maliciously or commits a theft, the DSO can revoke its permission to transact anymore. These actions are done through *getOrRevokePermission* function. By executing this function, *permit* variable of the transactor changes to either *true* or *false*. The power seller and buyer negotiate with each other in a forum outside the blockchain network. Once they both agree on a fixed price, the buyer submits the purchase request (*purchaseRequest*) to the contract which determines how much power and on what price he/she wants to buy from which transactor. After executing this function, the variable *purchase_Request* will be

equal to the negotiated price. Then, the seller submits the *trigger* which is a promise to sell power to a specific transactor at what price. A *Trigger* event is released to notify the DSO about this agreement. After that, the DSO can either accept or deny (*acceptOrDeny function*) the transaction based on the simulated effect of the transaction on the grid. Since performing this task in Ethereum Virtual Machine (EVM) is gas-consuming and every power transaction is inevitably associated with some amount of power losses, the DSO will demand some percentages of transactions from sellers to renew their permission. This percentage is not implemented in this contract because it depends on the policy of each DSO to how much and how often takes the money to renew the transaction permission. If a transactor refuses to respond to this demand, the DSO will terminate its permission.

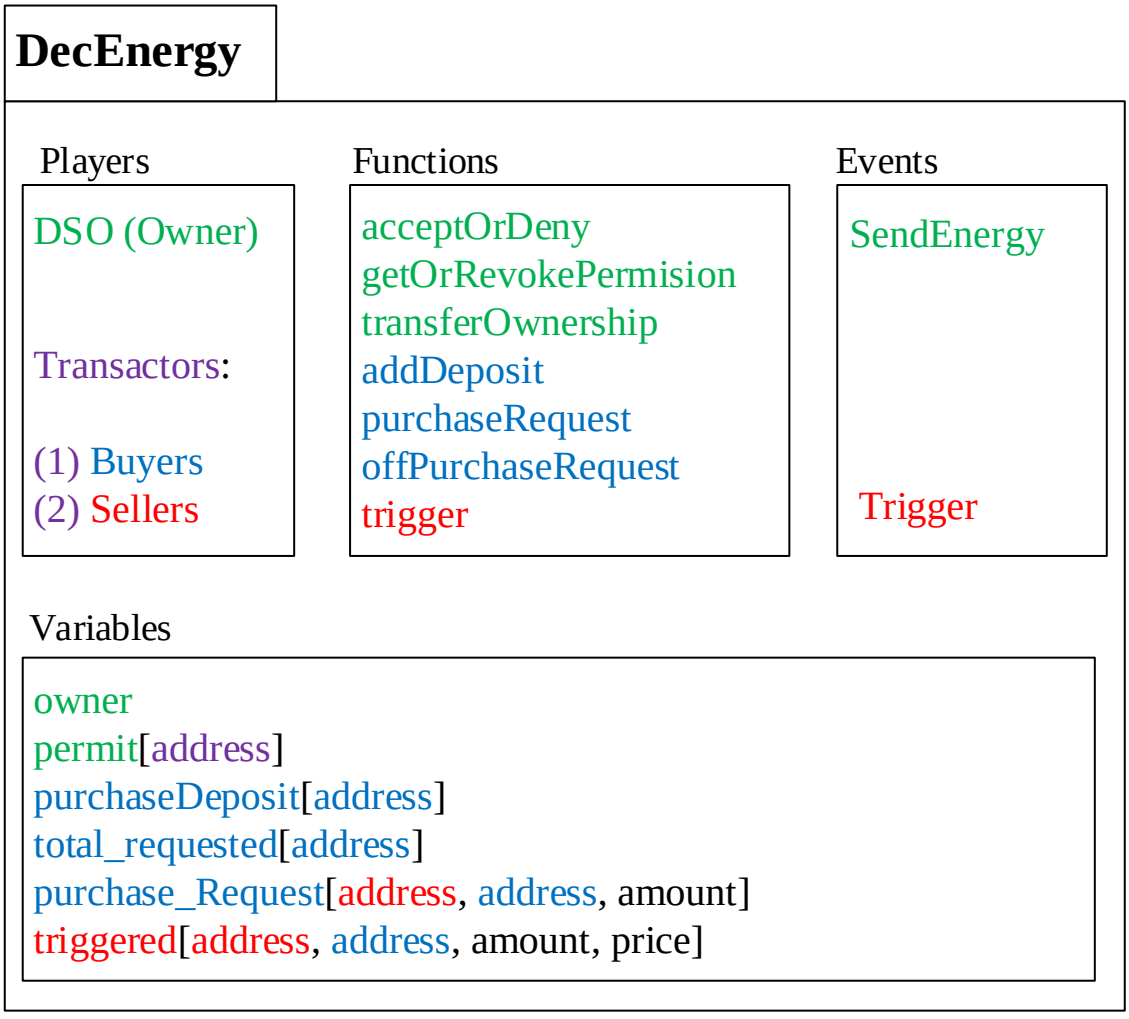

Fig. 1. The Overall Structure of the DecEnergyM smart contract