Blockchain Energy Market Place Evaluation: an Agent-Based Approach

Kei-Leo Brousmiche*, Andra Anoaica*, Omar Dib*, Tesnim Abdellatif†, Gilles Deleuze†
*IRT SystemX, Paris-Saclay, France, Email: {firstname.name}@irt-systemx.fr
†EDF R&D, Palaiseau, Email: {firstname.name}@edf.fr

***Abstract*—The growing interest for decentralized production of renewable energies calls for new market approaches. In this context, the blockchain is considered as a key technology for enabling decentralized local energy markets. Following this idea, several market solutions were proposed by the academic community. However, their technical feasibility or economical viability are yet to be assessed due to the lack of simulation frameworks. In this paper, we propose an agent-based simulation framework to experiment blockchain-backed energy market places. Moreover, based on realistic data of households located in northern France, we perform a sensitivity analysis to assess the impact of parameters on economics and blockchain system performances. Finally, we have implemented our solution on Raspberry Pies IoT devices to measure the actual power consumption of such systems.**

I. Introduction

The redistribution and pricing of renewable energy produced and exchanged among households within a community has become a research issue. Indeed, the growing interest of citizens toward decentralized production and political decisions have fostered green energies and accelerated the need for new market approaches. A classic approach to implement a market place is the use of a centralized system managed by a trust entity. While this kind of architecture is suited for trading applications, the cost could be an obstacle for local energy markets mainly for cost purpose. In this context, an increasing number of researchers and energy experts foresee the blockchain as a key technology to enable new cost effective local market places.

A blockchain can be defined as a trust-less, decentralized and continuously growing ledger of data blocks that have been validated by consensus among the participating nodes [1]. Each block contains timestamped data transactions, whose integrity and authenticity are guaranteed at any time thanks to hashing and public-key cryptographic algorithms. Once a new block is verified and written to the ledger, transactions cannot be altered retroactively without the collusion of the network majority. More recently, the emergence of smart contracts has unleashed the potential of blockchain in automating processes in a secure manner.

A smart contract is a digital protocol that verifies, executes and enforces contracts' terms that have been agreed between parties, without having to rely on third parties. Ethereum [2] is one of the first blockchain technology to have proposed a decentralized and turing-complete language, known as Solidity [3], to enable the development of smart contracts that offer the following properties: observability, verifiability, enforceability and security.

At a level of a local energy market place, the blockchain and smart contracts enable the trading of energy between the participating households through their blockchain nodes. The role of the auctioneer is represented by a smart contract that computes automatically the exchange prices and proceeds payments in a secure manner following a predefined allocation algorithm. Such systems has been recently proposed by several researchers [4], [5], [6], [7], [8]. However, few have implemented their solutions and moreover, technical feasibility, scalability and economical impact based on realistic data has never been assessed. Indeed, there is no framework available to experiment use cases with multiple, distributed participants interacting with a blockchain.

Agent-based modeling allows the simulation of autonomous entities interacting with their environment. This approach is thus suited to represent and study socio-economical phenomenon that emerges from a group of individuals [9]. Moreover, the distributed nature of the agents is adapted to a decentralized system architecture such as a blockchain.

In this paper, we propose an agent-based framework for simulating local energy market place integrating realistic consumption/production behavior and interacting with a private blockchain network. Based on data provided by the French energy utility company EDF describing the energy profiles of households located in the city of Lille, we experiment our system composed of 200 households agents, an Ethermint blockchain composed of 200 nodes and a smart contract implementing a double-auction allocation algorithm. In addition, we assesed the energy consumption of our system operated by a network of Raspberry Pies.

The paper is organized as follows: we first presents previous researches on the topic of blockchain-based local energy markets in section II. The simulation framework is described in section III. Section III-B describes the agent model of a household and its behavior. Based on simulations results, we analyze in IV the impact of multiple parameters, such as the proportion of consumers/prosumers, on the overall market. We also give some experimental results on the blockchain system performance and its energy consumption when running on IoT devices.

978-1-5386-7266-2/18/$31.00 ©2018 IEEE

II. Related Works

The use of blockchain in local energy markets addressed in the literature mainly focused on trading algorithms or money transfer mechanisms using cryptocurrencies [4], [5], [6], [7]. Moreover, the lack of technical evaluation of blockchain such as performance or scalability issues hinders their application within a real system.

For example, in [4] the micro-grid market is based on a continuous double auction (CDA) mechanism that matches buyers and sellers immediately upon the detection of compatible bids. The periodic double auction mechanism collects bids over a period of time and then clears the market at the expiration of a bidding interval. Agents in the CDA market compute the quotations based on adaptive aggressiveness (AA) strategies in order to achieve good profit. Analyses results are based on simulink simulations for a small community (8 consumers and 6 producers) and 100 market rounds. The blockchain system is used separately as a payment solution through digital currencies based on the Bitcoin plateform. These financial transactions are enabled when digital certificates of electricity transactions settlements are issued through software programs implementing off-chain trading market. Thus, a third trusted party is still needed to ensure some security and eventual contentious problems. Morevover, no technical evaluation has been conducted to guarantee that the Bitcoin blockchain is scalable or suitable for energy transaction settlements over time or efficient in a more realistic context.

In [6], the market is based on a double auction mechanism with a discrete closing time implementation with a smart contract on a private PoW Ethereum blockchain. Agents place orders in the market with a zero-intelligence (ZI) bidding strategy via blockchain accounts. As agents place bid orders, the money is locked-in until the settlement is carried out to ensure that every bid is sufficiently covered. Analysis results are based on simulation for 100 residential households with 15 min time slots. [8] proposed a similar solution but focuses on the simulation of the electric infrastructure and its energy loss. In both cases, the technological evaluation of the proposed blockchain solution has to be conducted in terms of computational resources, energy usage and transaction costs. Moreover, the PoW consensus protocol in private context isn't the suitable solution due to computational time in mining transactions. Thus, this may also influence on the solution scalability. Other architectures are proposed , like in [7], integrating the use of Smart Devices as advanced IoT and blockchain based solutions but no actual realization has been conducted yet.

We propose a blockchain implementation of a local energy market place based on a private blockchain network named Ethermint to tackle the system energy consumption issue. Moreover, we provide an agent-based simulation framework to experiment different market configurations and system scalability.

III. Technical Solution

This section describes our technical solution. It consists in a simulation framework and a local energy market place blockchain implementation. We first depict the architecture and relations between the different components of the solution. Then, we present the smart contracts representing the market place (III-A). Finally, we describe in detail the agent model and their behavior (III-B).

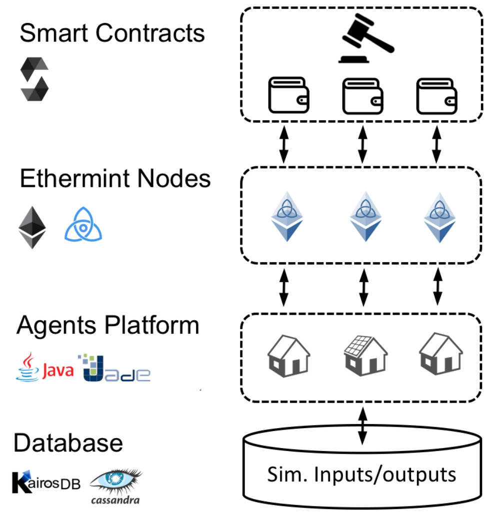

Fig. 1. Simulation Framework Architecture

As shown in Figure 1, the architecture has 4 layers.

a) Database: It stores the consumption and production profiles of 200 households based in the city of Lille during one week in summer and one week in winter on a time sampling of 1 minute. These data are provided by the EDF SMACH platform based on simulation[10]. It also records the outputs of the simulator (c.f. section IV). It has to be emphasized that this database component is here to: (1) facilitate data retrieval of electricity consumption/production for the agents and (2) store simulation outputs. Therefore, this centralized component does not hinder the distributed architecture of the upper layers and only serves as a simulation tool.

b) Agent Platform: Each agent simulates one household behavior defined by its energy production and consumption profiles. Based on these data, they propose to buy or sell energy volumes on the market at their own rates. The platform relies on JADE [11], a JAVA framework that enables the implementation of distributed multi-agent environment in which agents can run as threads on multiple hardwares. Thus, this platform enables to replicate real world asynchronous systems' challenges such as concurrency, deadlocks/livelocks and so on. Each agent has access to its local (i.e., on the same machine) blockchain node. The complete agent model and behavior are described in section III-B.

c) Ethermint Blockchain: This technology combines the Ethereum protocol with the Tendermint consensus engine [12]. Thus, it inherits all the capabilities of Ethereum, including the Ethereum Virtual Machine and the smart contracts support.

Moreover, by replacing the power and time consuming Eth-Hash consensus engine (i.e., the native engine of Ethereum) by Tendermint, it achieves fast transaction validation and low power consumption (see section IV for experimental results on the blockchain efficiency). Tendermint can be seen as a software for replicating an application on many machines in a secure and consistent manner. The protocol guarantees that machines compute the same state and can tolerate up to 1/3 of malicious or failing machines. Finally, writing ability is limited to a set of predefined validators nodes which retains attacks from outside of the network.

d) Smart Contracts: In our system we defined two smart contracts. The Wallet contracts represent the households' accounts and contain their Energy Token, the currency of the market place. Each agent has access to his own, personal Wallet. The MarketPlace contract represents both (i) the market place where the offers are gathered and (ii) the auctioneer who determines the market prices using a double auction mechanism at each market turn. These smart contracts are described in the following section.

A. Smart Contracts

This section details the MarketPlace and Wallet smart contracts, implemented in Solidity [3].

1) MarketPlace Contract: Deployed only once on the blockchain at the genesis of the blockchain, this contract embodies a *proposal* structure (similar to the concept of structure in C, it is a data model in Solidity) defined by a price in Energy Coin (a token used as the currency of the market), a volume (in kWh), a market turn index and the address of the proposer (i.e., the household's Wallet address). These proposals are stored in two tables, one for the bids and the other for the asks.

Two public functions are proposed to enable agents registering their proposals: $proposeBid$ and $proposeAsk$ both taking the volume of energy to trade, the desired unitary price expressed in EC (EC per kWh) and the target market turn that has to match the current turn. These functions register the proposal and trigger an allocation if the current market turn should be cleared. This market turn detection is done using the Solidity keyword now that return the current time (i.e. the timestamp of the last block). Since our Ethermint is cadenced at approximately 5 seconds, the accuracy of now is at this order of magnitude. The interval between two turns is a system parameter given in the genesis code. Also, this contract holds the prices of energy supply that can be modified in real time by authorized parties.

At the end of a turn, the contract executes a double-auction allocation algorithm. The details of the algorithm can be found in [13]. Basically, bids and asks are sorted in descending and ascending order respectively before being parsed to find a *critical point* where the demand volume meets the supply. Once the critical point is found, one selling price and one buying price are determined. All the proposals that have better prices than the critical point trade at these prices. This algorithm has the advantage of being strategic-proof, meaning that market manipulation is not possible.

2) Wallet Contract: One Wallet is deployed and owned by each household agent, it holds the EC detained by the household. This contract also stores the results of the last proposal outcomes: whether the demand has been satisfied or not, if positive, the volume of traded energy and the unit price.

B. Agent model

This section describes the agent model of a household and its behavior. These agents enables to simulate both the microscopic households behavior and the macroscopic community behavior once aggregated.

1) Data model: An agent represents a household of a community and can be modeled as a tuple:
$i = \langle id_i, creds_i, C_i, P_i, t_i, w_i, m, s_i \rangle$ with:

- $id_i \in \mathbb{N}$ the unique identifier of the agent
- $creds_i$ the Ethereum credentials composed of a private key and a public key that are required to authenticate the household agent when interacting with his blockchain node
- $C_i = \{c_t^s\}^s$ a list of energy consumption volumes $c \in \mathbb{N}$ measured in KWh corresponding to the time interval $t \in \mathbb{N}$ measured in minutes, containing $s \in \mathbb{N}$ elements (i.e., $s = 10080$ minutes or 7 days in the case of our use case)
- $P_i = \{p_t^s\}^s$ a list of energy production volumes as for C_i, empty if the agent i is not a prosumer
- w_i the blockchain address of the Wallet smart contract detained by the agent. It offers functions to: propose ask/sell offers to the market place, retrieve the amount of tokens held and retrieve the results of the last proposals (i.e., won/lost, traded volumes and prices)
- m the blockchain address of the unique MarketPlace contract which offers functions to retrieve the current market turn t_m and the prices at which an energy utility buys the energy B_t and sells the energy S_t in Energy Coins (EC)
- $s_i^t \in [min, max]$ with $min, max \in \mathbb{N}, min < max$ the duration in milliseconds of a time step t (agent's refresh rate) that is drawn randomly in a uniform manner at each time step

2) Behavior model: At each time step t, an agent i start by picking a random number s_i^t that cadences its execution rate: the agent will start the next time step after s_i^t milliseconds. Then it executes the following list of actions:

1 Retrieve the current market turn $t_m \in \mathbb{N}$ from the MarketPlace using his blockchain node and m (note that this is an ethCall and thus do not cost any gas unit)
2 If $t_i \geq t_m$, it means that the agent has already dealt his energy for the current turn. It restart the process after s_i^{t+1} milliseconds. Else, the agent has to deal with his energy and pursue the actions.
3 Update the step: $t_i := t_m$
4 Compute the energy surplus/lack: $\delta_i^t := p_i^t - c_i^t$
5 If $\delta_i^t = 0$, restart the process.

6 Send a bid proposal if $\delta_i^t > 0$ or an ask proposal if $\delta_i^t < 0$ using the functions $ProposeBid$ and $ProposeAsk$ of w_i. A bid is a tuple $i = \langle turn, type, volume, price\rangle$ with $turn = t_i$ the corresponding market turn, $type \in \{ask, bid\}$, $v \in \mathbb{N}$ the volume of energy proposed to be traded (i.e., $|\delta_i^t|$) and a $price \in [B_t, S_t]$. The interval of price is bounded since we make the assumption that agents are economically rational: consumers won't buy on the local market if the energy price is higher than the utility's selling price (upper bound S); conversely, producers won't sell their surplus to the local market if the utility buys it at a higher price (lower bound B). The price corresponds to one unit of energy (e.g., 130 Energy Coins for 1Wh). The agent picks randomly his price, accordingly to the type of proposal: $price \in [S_t, \frac{S_t+B_t}{2}]$ in case of bid, $price \in [\frac{S_t+B_t}{2}, B_t]$ in case of ask. This toy example of price selection is easily replaceable by other algorithms. However, as it will be shown in the experiments results, this zero knowledge strategy is sufficient to produce realistic price charts.

7 Wait for the MarketPlace response and write the results of the market turn into the database for the corresponding turn t_i: whether its offer has been granted, in the latter case the volume of traded energy (Wh), its unitary price (EC), the volume and price initially proposed.

8 Repeat from step 1.

IV. Experimentations

In the following section the economic benefits of the simulated market place will be estimated. As in any Market Place, we have offer and demand for a product and a currency used for payment. The product is energy locally produced and the demand is energy consumption. Offers are generated when there is a surplus of energy production coming from the households that are equipped with solar panels. The payment mechanism is enforced through tokens, which have an equivalent in fiat currency. In our simulation model, we assume that the energy utility sells its energy for 146 EC/kWh and buys the energy from the community for 120 EC/kWh, that corresponds to 1,46 euros and 1,21 euros. The houses in the neighborhood where the MarketPlace operates have interest to buy locally at a lower price than the energy supplier and to sell at a higher price. *This model and the prices settings are chosen for simulation purpose and are not linked to any pricing regulation or legal obligation context.*

To summarize, there are three main objectives of this MarketPlace:

- Minimize energy flows between a local community grid and a global grid (presented by a supplier in our example)
- Lower energy prices for the local consumers
- Reward local energy producers

In our experiments, we consider 200 households agents located in Lille during two weeks, one in summer (June) and the other in winter (January).

To present a complete picture of the market place, several possible scenarios will be tested whith two parameters: the period and the proportion of households in the neighborhood that both consume and produce energy (i.e. prosumers).

1) Input Data: The global energy consumption and production in the community is presented. Energy consumption as well as energy production are sensitive to the month. Production is function of the producers proportion in a community. Fig. 2 shows how the mean consumption and production variation in both periods. The consumption is higher in the winter period than the summer due to heaters use. We can also notice that the production quantity and its duration vary between the two seasons. In summer, days being longer, there is a prolonged exposure to sun of the solar panels which in turn generate more energy. This pattern will impact the market place behavior influencing prices and traded quantities.

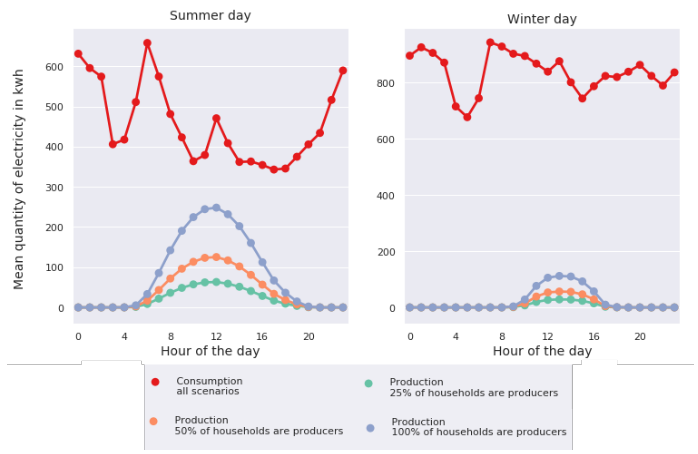

Fig. 2. Production and Consumption in Lille during a typical Summer and Winter day

2) Market behavior: Once a particular household produces more energy than it consumes, it sells the extra energy to the market place with a price that is more advantageous than the price offered by the energy supplier. In Figure 3 we present the quantity of energy traded in Lille in summer time on the Market Place, together with the mean gains in EC for the participants at each hour, with respect to the prices practiced by the energy supplier. We can see that as a bigger quantity is traded, buyers have to bid for higher prices to be able to buy energy making their gain in terms of tokens drop. On the other hand, we see that for producers gains are quite constant with values around 1EC.

Winter brings higher gains for the producers, while summer brings higher gains for the consumers. In the summer there is more energy to buy from the local market and higher offer brings smaller prices. On the contrary we observe an interesting phenomena on the producer side: the less there are producers on the market, the more they win. More producers means a higher concurrence and they have to lower the prices if they want their extra energy to be sold.

The market has winners and losers. More precisely, agents that wished to sell their energy at a higher price than the critical point, or consumers that wished to buy energy at lower prices than the price established by the marketplace. The

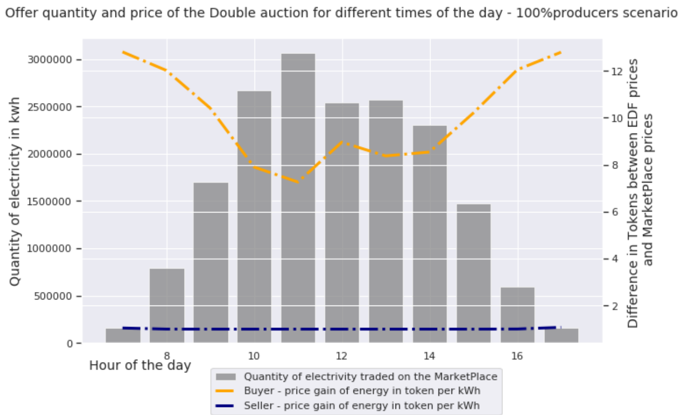

Fig. 3. Offer quantity and price of the Double auction for different times of the day - 100% producers scenario

Percentage of Producers	**Gains in Euro per day**		**Percentage of Overproduction Sold in the MarketPlace**	
	Summer	Winter	Summer	Winter
25%	0.54	0.08	71.66%	16.97%
50%	1.10	0.12	82.31%	24.32%
75%	2.14	0.29	88.47%	47.43%
100%	2.80	0.48	91.99%	60.00%

TABLE I
WEEKLY GAINS IN EUROS AND THE PERCENTAGE OF OVERPRODUCTION SOLD IN THE MARKETPLACE IN THE PROPOSED SCENARIOUS.

energy from these agents is purchased by the energy supplier that will pursue in distributing it to the neighborhood. Once the market has established who are the winners and the losers, those who did not manage to buy energy will receive it at the price offered by the energy supplier. From Table I, we can see an interesting phenomena concerning the proportion of overproduction sold. In scenarios where there is not much energy offered on the local market, namely winter period or when the proportion of producers is smaller, the percentage of energy sold is quite low. This highlights the idea that the Market Place is more efficient when the offer and demand volumes are larger.

Apart from minimizing the length of the energy that is produced, the Market Place also offers some financial gains for its participants. It is worth noting that participants can be both consumers and producers and the gains from each role need to be added. Table I presents the mean monetary gains per day for a household that only buys or buys and sells energy on the Market Place per day. In winter time due to small quantities transacted, the gains are insignificant.

A. Blockchain efficiency

In this section, we assess the performance of the Ethermint blockchain technology regarding its use within a real world distributed energy market place. As previously mentioned, this solution uses the Tendermint [14] consensus protocol that belongs to the family of BFT (Byzantine Fault Tolerance) algorithms, and inherits the capabilities of Ethereum including the EVM and smart contracts.

To make the assessment, several parameters are studied and various performance indicators are considered. The evaluation process consists of dynamically deploying a blockchain network on an Openstack virtual machine [15] (20 GB of RAM and 6 Virtual Central Processing Unit), and computing performance indicators. The versions used are Tendermint $0.12.0$ and Ethermint $0.5.3$.

Parameters used for building the blockchain are: the number of nodes n that represents the number of households in the community, and the number of validators (we assume here that each household is a validator node). Regarding the network topology, we consider that a complete graph is established between all nodes in the network.

Once the blockchain network is ready, a separate Java program run on a 8 GB of RAM is used in order to interact with the blockchain, as well as send transactions and compute performance indicators. As for all Ethereum based blockchains, instances of web3j client have been used. Moreover, this program will be in charge of sending transactions in asynchronous mode (*i.e.,* we do not wait for the transaction to be validated). The number of transactions to be sent in second as well as, the sent interval duration and the dynamic blockchain parameters are considered as input for this program. The sent transaction consists of calling the method $proposeBid$ in the MarketPlace Contract.

1) Results: We start with analyzing the impact of both the number of transactions sent per second, the number of validators with the number of transactions validated per second (Blockchain performance). The results are compared with the ideal case where all transactions sent are validated in one second or less. Figure 4 show that when the number of validators increases, the number of transactions validated per second decreases. This can be explained by the fact that more communications are required for the consensus to be accomplished. Besides, it has been noticed that the number of transactions sent per second has an impact on the output of the blockchain. More precisely, when more transactions sent per second, more the network accumulates some delays to validate transactions. For instance, in the worst case, when 10 transactions are sent per second, a network of 1, 2, 3 or 4 validators can almost validate them in one second or less. However, for a network containing 8, 12, 16 or 20 validators, the blockchain requires several seconds to validate all the input.

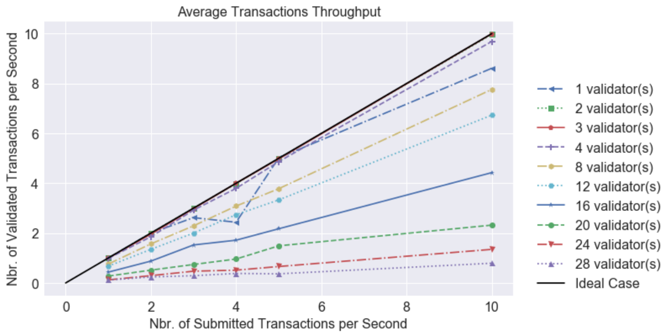

Fig. 4. Average transactions per second.

About the impact of the size of validators on the average transaction‘s validation time, Figure 5 shows that when the number of validators is important, the time required to validate a transaction increases also.

For instance, in a network containing 1 validator, the average validation time is very low (less than 1 second). However, it increases to approximately two minutes when 20 validators are considered. The confidence interval is also given in this figure.

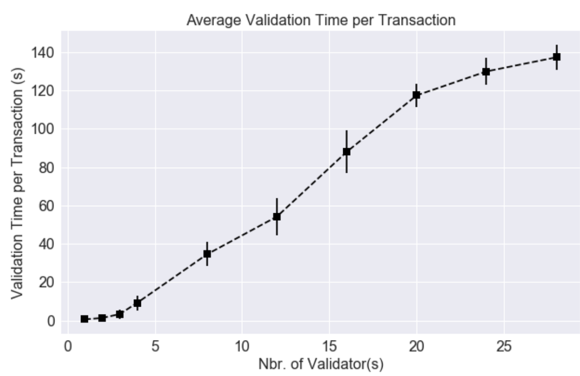

Fig. 5. Average transactions validation time (vs. number of validators).

From both Figures 4 and 5, it can be said that a compromise has to be made between the number of validators considered, the number of transactions sent per second and the expected blockchain performance. More precisely, an additional effort has to be made in order to select the appropriate number of validators in the network, and to fix the adequate number of transactions that the blockchain can deal with.

This usually depends on the use case where the blockchain is used. For instance, within a blockchain based energy market place, that requires validating 100 of energy exchange transactions every 20 seconds, it is obvious that it won't be possible to achieve that using more than 4 validators.

To deal with the energy market place, the number of transactions submitted per second is also an important parameter to study. Results from Figure 6 show that if the number of validators is fixed, the average validation time is slightly affected by the frequency at which transactions are submitted. This can be explained by the fact that Tendermint uses a memory pool to store all received transactions before validating them; the size of the pool does not really affect the rate at which Tendermint selects transactions and accomplishes its consensus mechanism. Besides, when the number of validators increases with the frequency of transactions, the average validation time drastically increases. Results indicate that up to 4 validators, the average validation time remains less than 10 seconds, however, it can increase up to 140 seconds when 24 validators are considered.

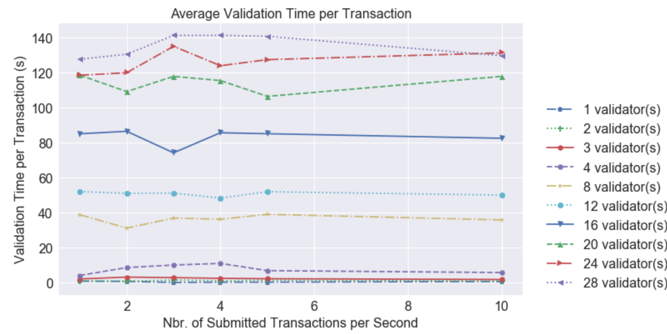

Fig. 6. Average transactions validation time (vs. number of transactions per second).

From these results, it can be concluded that the number of validators may constitute a bottleneck for the use of Ethermint as a blockchain technology to deal with a distributed Energy Market Place. That is, more validators implies high transactions' validation time, thus a restriction on the duration required to validate all energy exchange transactions between households.

B. Deployment on Raspberry Pies

Fig. 7. System running on 5 RPis blockchain network

To demonstrate the technical feasibility of deploying such system on IoT devices, we have deployed the proposed system on a network of 5 Raspberry Pies (RPi) as shown in Figure 7. We used the third generation of the RPi model B connected to one WiFi router. The operating system is Raspbian Strech, a Linux distribution adapted to RPi.

The Jade agent framework offers to deploy agents on a distributed architecture, thus we were able to deploy and run each household agent on the RPis without additional implementation.

The Ethermint blockchain nodes has been deployed manually on each RPi using an installation script.

We set the market turn interval to 15 seconds and measured the energy consumption of one RPi during 10 minutes in Figure 8. The resulting consumption's average is 1,58 Watt (or VA) with several peaks at more than 2W. We also measured that after one hour of execution, the cumulated consumption is 0.304Wh.

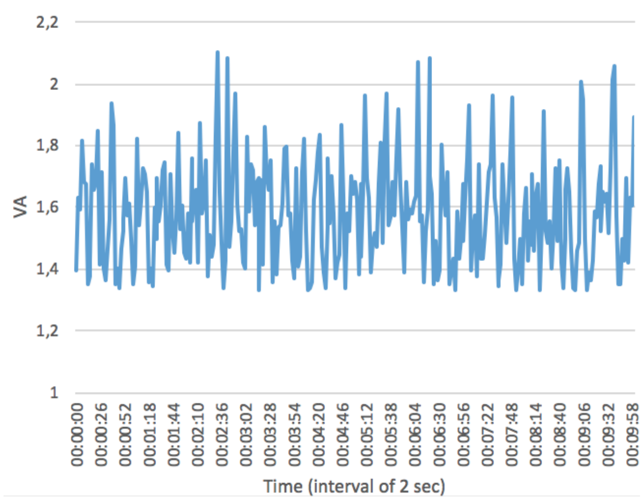

Fig. 8. Energy consumption of a Raspberry Pie participating to the market place

V. Conclusions and Perspectives

In this paper, we proposed a framework enabling the simulation of autonomous agents interacting with a blockchain. We implemented an energy market place based on a smart contract executing a double auction mechanism which enables the trading of locally produced energy. Based on realistic data of 200 households' energy profiles, we studied the influence of producers proportion in the community with respect to the economical gains. The technical feasibility and impacts of the system parameters on the blockchain efficiency have also been assessed. Finally, we deployed the system on a network of Raspberry Pies to demonstrate that the proposed solution is achievable on IoT devices.

In future works, we intend to study experimental results of this market place on real size test houses. Also, other allocation algorithms have to be studied to evaluate their impact on the economical outcomes. Finally, we are currently developing cryptographic protocols to ensure data privacy of information transiting on the blockchain network.

Acknowledgment

This research work has been carried out under the leadership of the Institute for Technological Research SystemX, and therefore granted with public funds within the scope of the French Program Investissements d'Avenir.

References

[1] E. B. Hamida, K. L. Brousmiche, H. Levard, and E. Thea, "Blockchain for enterprise: Overview, opportunities and challenges," *ICWMC 2017*, p. 91, 2017.

[2] "Ethereum project," accessed: 2018-07-06. [Online]. Available: https://www.ethereum.org/

[3] "Solidity: a contract-oriented, high-level language for implementing smart contracts," accessed: 2018-07-06. [Online]. Available: http://solidity.readthedocs.io

[4] J. Wang, Q. Wang, N. Zhou, and Y. Chi, "A novel electricity transaction mode of microgrids based on blockchain and continuous double auction," *Energies*, vol. 10, no. 12, p. 1971, 2017.

[5] A. Abidin, A. Aly, S. Cleemput, and M. A. Mustafa, "Secure and privacy-friendly local electricity trading and billing in smart grid," *arXiv preprint arXiv:1801.08354*, 2018.

[6] N. B. B. C. e. a. Mengelkamp, E., "A blockchain-based smart grid: towards sustainable local energy markets." Springer Berlin Heidelberg, 2018.

[7] F. Lombardi, L. Aniello, S. De Angelis, A. Margheri, and V. Sassone, "A blockchain-based infrastructure for reliable and cost-effective iot-aided smart grids," 2018.

[8] J. Horta, D. Kofman, D. Menga, and A. Silva, "Novel market approach for locally balancing renewable energy production and flexible demand," Dresden, Germany, 2017.

[9] D. Helbing, "Agent-based modeling," in *Social self-organization*. Springer, 2012, pp. 25–70.

[10] E. Amouroux, T. Huraux, F. Sempé, N. Sabouret, and Y. Haradji, "Simulating human activities to investigate household energy consumption." in *ICAART (2)*, 2013, pp. 71–80. [Online]. Available: http://perso.limsi.fr/sabouret/ps/icaart2013.pdf

[11] F. Bellifemine, F. Bergenti, G. Caire, and A. Poggi, "Jade-a java agent development framework," in *Multi-Agent Programming*. Springer, 2005, pp. 125–147.

[12] J. Kwon, "Tendermint: Consensus without mining," *URL http://tendermint. com/docs/tendermint {_} v04. pdf*, 2014. [Online]. Available: https://cdn.relayto.com/media/files/LPgoWO18TCeMIggJVakt_tendermint.pdf

[13] P. Huang, A. Scheller-Wolf, and K. Sycara, "A strategy-proof multiunit double auction mechanism," in *Proceedings of the first international joint conference on Autonomous agents and multiagent systems: part 1*. ACM, 2002, pp. 166–167.

[14] E. Buchman, "Tendermint: Byzantine fault tolerance in the age of blockchains," Ph.D. dissertation, 2016, accessed: 2018-05-25. [Online]. Available: https://allquantor.at/blockchainbib/pdf/buchman2016tendermint.pdf

[15] OpenStack. (2018) OpenStack description. Accessed: 2018-05-25. [Online]. Available: https://www.openstack.org/