Demo: EVA: Fair and Auditable Electric Vehicle Charging Service using Blockchain

Jelena Pajic[1,2], José Rivera[1,2], Kaiwen Zhang[1,2,3,4], Hans-Arno Jacobsen[1,2,3]
Technical University of Munich
Middleware Systems Research Group
University of Toronto
École de technologie supérieure

ABSTRACT

The recent success of electric vehicles leads to unprecedentedly high peaks of demand on the electric grid at the times when most people charge their cars. In order to avoid unreasonably rising costs due to inefficient utilization of the electricity infrastructure, we propose EVA: a scheduling system to solve the valley filling problem by distributing the electricity demand generated by electric vehicles in a geographically limited area efficiently over time spans in which the electric grid is underutilized. EVA is based on a smart contract running on the Ethereum blockchain in combination with off-chain computational nodes performing the schedule calculation using the Alternating Direction Method of Multipliers (ADMM). This allows for a high degree of transparency and verifiability in the scheduling computation results while maintaining a reasonable level of efficiency. In order to interact with the scheduling system, we developed a decentralized app with a graphical frontend, where the user can enter vehicle information and future energy requirements as well as review upcoming schedules. The calculation of the schedule is performed on a daily basis, continuously providing schedules for participating users for the following day.

CCS CONCEPTS

• **Computer systems organization** → **Distributed architectures**; *Client-server architectures*; *Peer-to-peer architectures*; • **Software and its engineering** → Distributed systems organizing principles;

KEYWORDS

Electric Vehicles, Blockchain, Ethereum, Smart Contract, Energy Systems, Electric Grids, Scheduling

ACM Reference Format:
Jelena Pajic[1,2], José Rivera[1,2], Kaiwen Zhang[1,2,3,4], Hans-Arno Jacobsen[1,2,3]. 2018. Demo: EVA: Fair and Auditable Electric Vehicle Charging Service using Blockchain. In *DEBS '18: The 12th ACM International Conference on Distributed and Event-based Systems, June 25–29, 2018, Hamilton, New Zealand.* ACM, New York, NY, USA, 4 pages. https://doi.org/10.1145/3210284.3219776

DEBS '18, June 25–29, 2018, Hamilton, New Zealand

ACM ISBN 978-1-4503-5782-1/18/06.
https://doi.org/10.1145/3210284.3219776

1 INTRODUCTION

The number of privately-owned electric vehicles (EVs) on the road keeps growing at a significant pace [4]. While this has numerous beneficial implications such as reduced noise levels, lower greenhouse gas emissions, and vastly improved energy efficiency in general, there is also a major drawback on the common infrastructure: unprecedented high peaks of load on the electric grid [6]. This effect is caused by two factors: electric vehicles have high energy demands in order to charge in a timely manner [3] and most people charge their vehicles at similar and therefore overlapping times - after work when they reach their home [3].

This leads to very high demand on the electric grid at specific times in areas where many electric vehicle owners live close together. Compared to this demand, the average load on the electric grid is relatively low. The current situation therefore implies two equally unfavorable consequences:

- The common electricity production infrastructure must be sized to meet this peak demand, leading to significant underutilization during off-peak times, or
- Electricity providers are forced to equalize the deficit in available electricity by buying from other, remote suppliers using expensive *intra-day trading* [5].

Both options incur a significant amount of avoidable costs that would have to be shouldered by society in general in form of a considerable rise in energy prices in the long term.

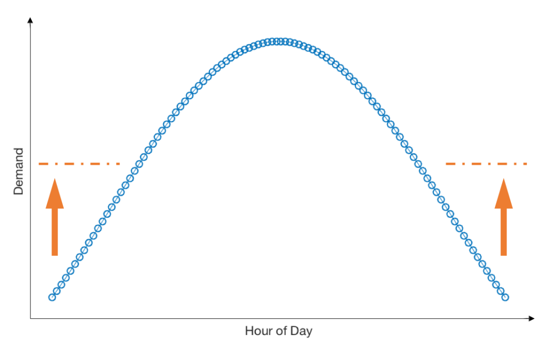

Figure 1: Valley Filling

One of the possible solutions to these issues is referred to as valley filling scenario, as seen in Figure 1. It focuses on shifting the EV load away from the system peak into evening hours when the load is low and the network's capacity is high. This strategy offers the possibility to take advantage of low electricity prices and an increased network capacity during night time as well as of avoiding additional load peaks. Desired charging behavior is imposed by the aggregator, charging at peak load times is avoided and shifted to low load hours.

Since the proposed solution involves sophisticated computations (such as ADMM) and includes privately accessible data (such as current electricity intra-day trading rates), users could find it challenging to trust the results and value their individual contribution. In order to increase the overall trust in the scheduling system, we employ blockchain technology to make input data as well as every computational result publicly accessible, immutable and verifiable.

The contributions of this paper are:

(1) We present EVA, an EV charging scheduler application, built using smart contracts on Ethereum.
(2) We leverage the use of blockchain to provide transparency and verifiability of the computed results.
(3) We provide a graphical frontend for customers to register and query information via the blockchain.

2 BACKGROUND ON BLOCKCHAINS

A blockchain-based distributed ledger records transactions, which are immutable once committed to the chain. The use of a blockchain platform allows its users to no longer have to trust a centralized authority, instead relying on the underlying cryptography mechanisms. Thus, blockchain-based systems meet the need for transparency and security in an cost-effective manner.

Generally, there are two types of blockchains: public and permissioned (sometimes called private). The key difference is that everyone can participate in the maintenance of a public blockchain (e.g. mining), while permissioned blockchains require authorization by a governing authority [2].

Blockchain was originally conceptualized and implemented for cryptocurrency [7]. However, the introduction of smart contracts created a growing number of application areas, e.g. electric vehicles, Internet of Things, health care, smart property etc.

Smart contracts are arbitrary code on blockchain which automatically implement terms of multi-party agreements [1]. They are executed by a network of mutually distrusting nodes, allowing to automate and track certain state transitions.

As of this writing, the most widely used public blockchain that supports smart contracts is *Ethereum*. It currently uses Proof-of-Work based on the *Ethash* algorithm as consensus algorithm and employs the *Ethereum Virtual Machine (EVM)* to execute the compiled bytecode of smart contracts on every full node in the network. The most popular human-readable programming language for the development of smart contracts in the Ethereum ecosystem is *Solidity*.

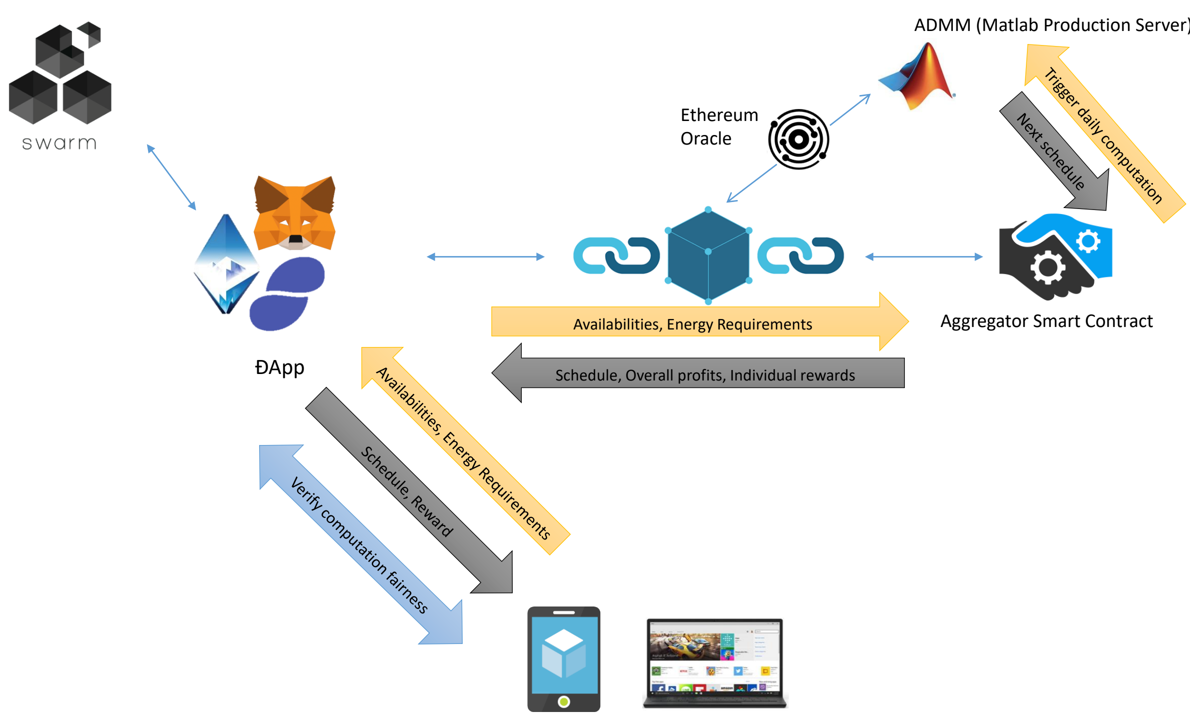

Figure 2: EVA Architecture Overview

3 SOLUTION MODEL

We propose a solution which allows electric vehicle owners to provide flexible charging times, which are used by electricity providers to schedule the charging process efficiently across multiple EVs in order to reduce the overhead on the electric grid. In turn, participating owners will be rewarded proportionally for their contribution to the reduction of costs that would otherwise be incurred by inefficient utilization of the electric grid infrastructure or the use of intra-day trading. The goal of the scheduling is to find a reasonable solution to the valley filling problem by distributing the peak demand efficiently onto time slots where the infrastructure is underutilized [8].

In EVA, electric vehicle owners use their computers or smartphones to register their EV within a certain geographic area together with its general energy requirements (e.g. battery capacity) and their flexibility concerning charging times for specified time frames. The requirements and flexibility entered can be given for any timespan in the future, but should be submitted at least one day in advance to be considered in the next scheduling cycle.

The given inputs are persisted in a smart contract, called *Aggregator*, on a blockchain, where the relevant data for the following cycle is read every 24 hours by a scheduling system, which computes an approximately optimal distribution by dividing every given 24 hour period into 96 equally-sized time slots and iteratively applying a convex optimization algorithm called Alternating Direction Method of Multipliers (ADMM)[8]. One scheduling instance should be deployed per geographical location that is dependent on a defined subsection of the electric grid and subject to a given amount of load generated by electric vehicles. The resulting schedule is persisted on the blockchain.

After generating the schedule, the *Aggregator* computes individual rewards for every participating EV owner proportional to their contribution to the amount of costs saved. The contribution is measured based on the time flexibility given relatively to the required amount of electricity. Together with the schedule, the overall profit as well as the individual profits for each participant are stored on the blockchain as shown in Figure 2.

Since every input and every result is available on a public blockchain and immutable and accessible by everyone, each participant has the possibility to verify the fairness of the profit distribution.

4 ARCHITECTURE AND IMPLEMENTATION

For our prototype implementation, we decided to develop the *Aggregator* smart contract in Solidity and deploy it on the Ethereum blockchain. This decision is based on the fact that Ethereum is the most widely used public blockchain supporting all of our requirements for smart contract development. The deployment over a public blockchain is desirable since the service should be open to an unrestricted user group and the proposed idea of trust relies on immutability and free accessibility of all computational data. An overview of all involved system components can be seen in Figure 2.

As the computation of the schedule itself is quite costly, it is currently impossible to run the entire system on the blockchain. Therefore, we decided to deploy separate computational nodes to run the actual ADMM algorithm. These nodes are represented on the blockchain in form of an Ethereum *Oracle*, taking the input data and writing back the results using blockchain transactions as shown in Figure 2. The calculation of the schedule is triggered automatically at the same time every day, providing the schedule for the following 24 hour period.

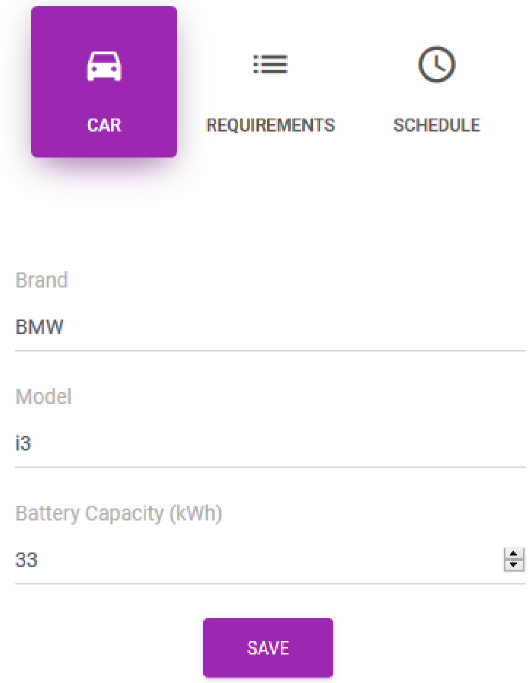

Figure 3: User interface

To provide a convenient interface for user access, we developed an Ethereum Distributed Application (ÐApp) based on HTML, CSS, Javascript and the web3.js library provided by the Ethereum project. The web3.js library is necessary for asynchronous calls to the blockchain in order to interact with the *Aggregator* smart contract. For this prototype, we decided to rely on the Infura project to provide us with publicly accessible Ethereum nodes to reduce the barrier of entry for possible users. Therefore, EVA is usable through any ÐApp browser for Mobile and Desktop Platforms currently available in the Ethereum Ecosystem such as MetaMask, Mist, or Status. A screenshot of the interface can be seen in Figure 3.

5 DEMONSTRATION

To demonstrate the functionality of the scheduling system, we have prepared a concise walkthrough that presents the core features of the prototype.

(1) The user opens the EVA frontend using an Ethereum ÐApp browser and registers an electronic vehicle using the respective view by entering make, model and battery capacity of the car. This step requires a transaction on the Ethereum blockchain that incurs a minor cost in form of *gas* (a fee paid to the miners for processing transactions) and has to be approved by the user.

(2) After the transaction is confirmed by the network, the participant switches to the energy requirements view and adds at least one requirement in form of a timespan and requested electricity for a chosen date. Similar to before, this step also requires a transaction and therefore incurs a minor gas cost.
(3) The schedule calculation is triggered on the Matlab instance and the resulting schedule is sent to the blockchain.
(4) As soon as the transaction from the previous step is confirmed, the user can switch to the schedule view and review the assigned charging timeslot.
(5) After the charging is complete, it is possible to query the blockchain for the corresponding reward.
(6) The user can instruct EVA to verify the fairness of the reward distribution.

REFERENCES

[1] V Buterin. 2016. Ethereum white paper: A next-generation smart contract and decentralized application platform. First version (2014).
[2] Christian Cachin. 2016. Architecture of the Hyperledger blockchain fabric. In *Workshop on Distributed Cryptocurrencies and Consensus Ledgers*.
[3] Kristien Clement-Nyns, Edwin Haesen, and Johan Driesen. 2010. The impact of charging plug-in hybrid electric vehicles on a residential distribution grid. *IEEE Transactions on power systems* 25, 1 (2010), 371–380.
[4] Till Gnann, Patrick Plötz, André Kühn, and Martin Wietschel. 2015. Modelling market diffusion of electric vehicles with real world driving data–German market and policy options. *Transportation Research Part A: Policy and Practice* 77 (2015), 95–112.
[5] Rüdiger Kiesel and Florentina Paraschiv. 2017. Econometric analysis of 15-minute intraday electricity prices. *Energy Economics* 64 (2017), 77–90.
[6] João A Peças Lopes, Filipe Joel Soares, and Pedro M Rocha Almeida. 2011. Integration of electric vehicles in the electric power system. *Proc. IEEE* 99, 1 (2011), 168–183.
[7] Satoshi Nakamoto. 2008. Bitcoin: A peer-to-peer electronic cash system. (2008).
[8] Jose Rivera, Philipp Wolfrum, Sandra Hirche, Christoph Goebel, and Hans-Arno Jacobsen. 2013. Alternating direction method of multipliers for decentralized electric vehicle charging control. In *Decision and Control (CDC), 2013 IEEE 52nd Annual Conference on*. IEEE, 6960–6965.