Blockchain Based Autonomous Selection of Electric Vehicle Charging Station

Matevž Pustišek
Faculty of Electrical Engineering,
University of Ljubljana,
Ljubljana, Slovenia
matevz.pustisek@fe.uni-lj.si

Andrej Kos
Faculty of Electrical Engineering,
University of Ljubljana,
Ljubljana, Slovenia
andrej.kos@fe.uni-lj.si

Urban Sedlar
Faculty of Electrical Engineering,
University of Ljubljana,
Ljubljana, Slovenia
urban.sedlar@fe.uni-lj.si

***Abstract*— This paper introduces a concept of autonomous blockchain based negotiation to select the most convenient electric vehicle charging station. The enabling blockchain technologies are well known fundament of cryptocurrencies, but offer many other possible applications areas, such as automated trusted machine-to-machine transactions, including auctions, bidding and payments. Compared to traditional centralized approaches, such a solution does not require any central entities and can be fully automated, including the payment for the energy. Based on e.g. the planned route, car battery status, real-time traffic information and drivers' preferences, a car could request charging bids from various charging stations along the route, by executing blockchain based smart contracts related to these charging stations. It would then select the most appropriate one based on offered prices, but other input parameters could be also taken into account, e.g. waiting times, estimated charging duration and alike. The smart contract execution could be extended to reservation and payment, too. In the paper we briefly present blockchain technologies, with the focus on Ethereum, and explain the role of smart contracts. We outline the architecture of a simple system for autonomous selection for electric vehicle charging station and provide a UML model to depict the activities of the actors involved in these operations and to clarify the role and the requirements of various blockchain related entities. The aim of our research paper is to outline a possible use-case for a blockchain prototype implementation. The implementation, which is not presented here, will serve for investigation of practical aspects of smart node development and operation, like transaction performance measurements, practical system requirement evaluations and comparison of reliability of various Ethereum clients. By our research we hope to support the investigation of novel applications and blending Internet of things with blockchain technologies, and in particular for elaboration of possible new use cases in the domain of electrical energy and mobility.**

Keywords— blockchain, smart contract, electric vehicle, charging station, autonomous negotiation, UML

I. Introduction

Internet of things (IoT) the well-established concept of numerous interconnected *things* along with corresponding cloud or fog based applications is revolutionizing the Internet and is already being deployed in various applications domains.

IoT reflects also in recent developments in automotive and energy supply sectors with e.g. autonomous or electric vehicles (EV), and smart grid (SG). A modern car is not only operated by numerous in-car computer and communication system. A car has become an internet device, with pervasive internet connectivity to support safety (automated emergency calls), diagnostics, and entertainment [1]. The shares of full-battery and plugin-hybrid EV might not be high at the moment, but the raise is impressive [2], [3]. At the same time infrastructure is being built in many countries, including electric charging stations (CS), so electric mobility seems to be obvious future. Smart grid is response to new challenges that electric supply grids are facing. These include e.g. energy production in distributed power sources, smart

978-1-5090-5952-2/16 $31.00 © 2016 IEEE

DOI 10.1109/IIKI.2016.60

metering and demand side management, electric vehicles, or energy storage, along with the indisputable requirements for reliable and sustainable operation of electric supply network.

A seemingly unrelated concept, which is younger and potentially even more disruptive that IoT, are the blockchains (BC) technologies. They can be explained as a trusted communication and computation platform, which requires no central trust entity to assure transparent and irrevocable electronic transactions among persons or machines. Enabling technologies of BC are sophisticated encryption mechanisms and peer-to-peer communications. The blockchain technologies are well known fundament of cryptocurrencies (e.g. Bitcoin), but offer many other possible applications areas. For our research automated trusted machine-to-machine transactions, including auctions, bidding and payments are of special interest, because they could be applied in IoT, including EV and SG solutions.

A combination of IoT and BC enables a variety of novel internet applications, not only in technical, but even more importantly in new business aspects. The objective of our paper is to indicate, that an EV charging system is possible, where application of BC enables a fully autonomous selection of CS, including negotiation for the most convenient option, and automated payment for the charging. In Section 2 we briefly outline the role and operation of BC technologies, in Section 3 we develop a model of autonomous selection of CSs based on BC, and in Section 4 we elaborate open issues regarding the proposed solution.

The aim of this paper is thus to outline a possible use-case for a blockchain prototype, which we plan to implement next. We will implement it on Raspberry Pi and raspbian operating system. The objectives of this foreseen implementation are to analyze and verify the available Ethereum clients for this environment, to evaluate practical system requirements for full and light clients, to measure transaction throughput, and to investigate impact of public and private blockchain network implementations. By our research we hope to support the investigation of novel applications and blending Internet of things with blockchain technologies, in particular for elaboration of possible new use cases in the domain of electrical energy and mobility.

II. BLOCKCHAINS FOR IOT

The two key reasons that make blockchains so appealing to IoT are scalability of IoT backend and data monetization. Despite the continuous growth of cloud capabilities, its' predominantly hub-and-spoke oriented architecture might not be able cope with the expected number of IoT devices. Additional limiting factor in this could be the capacity of internet connections towards the cloud backends [4]. Regarding monetization, blockchains enable low cost machine-to-machine micro transaction. Costs of such transactions are minimal, especially compared to costs in current e-payment systems. In this way an arbitrary small market for available electronic data can be created.

In terms of IoT a blockchain can be seen as a distributed and trusted database, where the act of inserting/reading a parameter value is called a transaction, which is verified by a distributed community. The blockchain technology does not (necessarily) provide privacy of data. Therefore new trusted and private implementations of blockchain systems are being investigated [5]. A set of new transactions from blockchain clients is organized in a block, a structure which is then cryptographically verified by a network of distributed verifiers in a blockchain community. A newly verified block is in this way added to the chain of blocks of previous transactions. In this way the record of transaction is permanently stored in a public digital ledger, and cannot be removed, changed or retracted. There is no central entity. The system is fully distributed, which makes it robust and scalable even beyond any cloud based data storage service. In this way the blockchain builds a secure transaction layer and a framework of digital assets (electronic data that cannot be duplicated), which can be moved or sold from one proprietor to another. Various specifications and implementations of blockchain technologies are available, but two most prominent are gathered around the Bitcoin [6] blockchain protocol and around Ethereum [7]. A specific blockchain protocol can have many implementations, and can thus build various frameworks with corresponding communities of users and application emphases. Key implementation of Bitcoin is crypto value system bitcoin [8]. Ethereum network on the other hand, is more oriented towards creation of distributed applications, although a cryptocurrency (the ether) is also available.

A. Ethereum

The Ethereum white paper [9], which is the initial document describing Ethereum, explains that *the Ethereum protocol was originally conceived as an upgraded version of a cryptocurrency, providing advanced features such as on-blockchain escrow, withdrawal limits, financial contracts, gambling markets and the like via a highly generalized programming language. The Ethereum protocol would not "support" any of the applications directly, but the existence of a Turing-complete programming language means that arbitrary contracts can theoretically be created for any transaction type or application. What is more interesting about Ethereum, however, is that the Ethereum protocol moves far beyond just currency.* The Ethereum network of course includes its own currency. This built-in currency, *ether, serves the dual purpose of providing a primary liquidity layer to allow for efficient exchange between various types of digital assets and, more importantly, of providing a mechanism for paying transaction fees.* For these reasons Ethereum emerged as the platform for (i) financial (currencies, token systems), (ii) semi-financial (e.g. crowd sensing) and (iii) non-financial applications (on-line voting, decentralized governance). Besides distributed storage for IoT, Ethereum can easy facilitate decentralized data feed [9], where a vast amount of numerous concurrent (low fidelity) measurements is summarized into e.g. the most possible value of temperature.

The nodes – the entities of Ethereum network – can participate in one of two networks, the mainnet or the testnet. As names indicate the testnet is meant for testing purposes and all value transfers operate with a counterfeit cryptocurrency, which cannot be exchanged for other cryptocurrencies or real money. Each node can manage several Ethereum accounts i.e. objects which are identified by a unique addresses and protected by sophisticated cryptographic mechanisms. There are two types of accounts in Ethereum [10]. The externally owned accounts are controlled by private keys, which are generated during the account creation and are known only to the account owner. The account owner can then create transactions from such an account. The contract accounts are controlled by their contract code. The code is executed every time such an account receives a message/transaction from another account. The interaction between the accounts is fully decentralized. A discovery mechanism based on kademlia peer-to-peer protocol is used, with several bootstrap nodes that maintain the lists of regular nodes.

B. Distributed applications in Ethereum

The contracts in Ethereum are not some formal requirements or obligations, but can be more adequately explained as autonomous agents, whose behavior is determined by their contract code. This code is executed every time this account receives a message, which is a transaction addressed to it.

Ethereum account address is a 160 bit number. When a new account is created, a random private key is generated and a public key is derived from it. A 256 bit hash is calculated form the private key and last 160 bits of this hash are the address [12]. Besides the address, an account is characterized by its' current ether balance, and optional contract code and storage [9]. The contract accounts with the embedded contract code have a direct control over their own ether balance and their own key/value store. Transactions in Ethereum enable transfer of value and information between accounts.

To develop a smart contracts and thus a distributed applications a computationally universal (i.e. Turing complete) language is provided. Such a language needs some form of conditional repetitions or conditional jumps (e.g. while loops, if commands) and means to read and write to some form of storage. In Ethereum a contract code has access to three different types of space to store data, including stack, memory and long-term store for key-value pairs.

The fundamental language is the low-level bytecode language and Ethereum provides a virtual machine (i.e. Ethereum virtual machine, EVM) which executes such code. The code can also access the ether balance, sender address and data of the incoming message, as well as block header data, and the code can also return a byte array of data as an output [10]. Several high(er) level languages are available for application development. The current flagship is Solidity [11] – JavaScript like language, but other languages have been used in the past, e.g. LLL and Serpent. Higher level code is compiled to bytecode prior to execution in EVM.

C. Solidity smart contracts

In Solidity smart contract lifecycle consists of writing (programming), compiling to bytecode and placing it on the chain. From this point one, it can receive messages addressed to its' account address. The contract code is run when the account receives a message [10]. A message contains sender (implicit) and recipient address, amount of ether to be transferred, optional data field for input parameters and maximum amount of gas. A message is processed by the blockchain in the same way as a transaction.

A simple way to try the contract development is the online compiler [12] or by using a wallet (e.g. Mist) which compiles a Solidity contracts and stores the bytecode to the chain. A Solidity contract code comprises of contract declaration along with optional contract constructor function, which is called only upon the creation of the contract. In contract code there are declarations of state variables, which can be set as public to be accessible by other contracts, declaration of functions and events.

III. EV CHARGING SELECTION WITH BLOCKCHAINS

There are many challenges in power grid transformation, ranging from distributed energy resources and energy storage, to reliability, efficiency and management of the grid, and enabling the foreseen large-scale integration of plug-in electric vehicles (PEVs) [13]. Automation of these processes requires novel machine-to-machine communication paradigms to interconnect energy producers, providers and consumers. The idea of applying blockchain technologies in IoT application domains related to (electrical) energy provisioning appeared soon after realizing the potentials of crypto currencies and smart contracts in for IoT. Nevertheless, most of the initiatives are still in very early stages. Some examples include real-time metering of local energy generation and use [14] and bitcoin prepaid smart energy metes [15]. Blockchain transactions are considered as the fundament for automated energy trading applications. Bitcoin is applied in a pilot project for one time electricity charging, including of EV [16], [17].

EV charging infrastructure are being rapidly built all around the world. Nevertheless using the system of private or public CS is currently not yet a common and clearly defined experience. First the user has to select a CS from one of the directories, managed by CS operators or by independent and dedicated social networks with large user communities and (potentially) global coverage (e.g. Plugshare [18]). These directories provide information about public and residential CS locations, as well as about tariffs, charging power and plug types, etc. Plugshare e.g. has option of user comments about particular CS and their experience, charging reservation system and trip planner. The tariffs charging models vary a lot, too. It is not uncommon, that you can obtain free charging. When charging is not free of cost, it can be flat rate (by visit), by the amount of charged energy, or by time, spent at the CS. Charging costs are often substantially higher along the highways. The charging requires user (car) authentication and payment. This is usually done with a combination of dedicated RFID cards, issued by CS operators or energy providers. Use of this authentication is associated with registration in a corresponding mobile or web applications. Payment is executed with common e-payment systems.

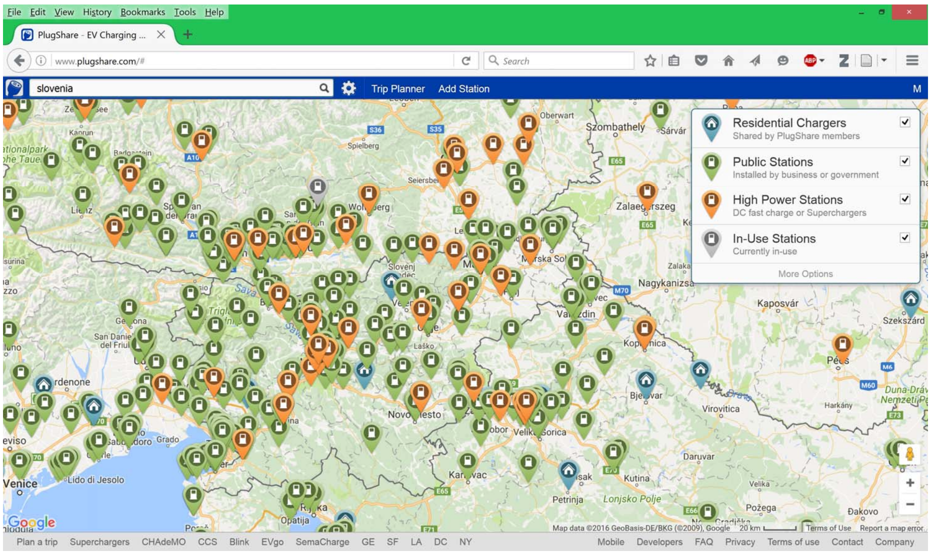

Figure 1: Directory of EVCSs in Slovenia [21]

Apart from the fact that EV charging approaches are not unified in technical terms and are diverse in business models, there are several other issues that might prove to limiting in successful large-scale adoption of EVs and corresponding charging mechanisms. We believe that BC based approach is promising for following reasons: (i) no trusted central entity (for e.g. authentication and billing) is needed, (ii) no subscription or prior business arrangement is required, (iii) transparent service negotiation/agreement that includes instant payment for charging can be achieved, and (iv) full automation of the session is possible.

A. System outline

Disregarding the implementation approach (manual, automated, real time, etc.), following activities are related to EV charging: (i) selection of the CS, (ii) timeslot reservation (optional), (iii) user/vehicle authentication at the CS, (iv) charging and (v) payment. In our proposal we elaborate only the (i) selection of CS. But the same approach can be extended to (ii) and (v), too. The (iii) authentication is assured by the blockchain mechanisms by default.

In the proposed solution there are three actors foreseen: a driver, an electric vehicle and a charging station. The driver has a simple function of launching the selection procedure and finally confirming the bid suggested by the car. The car is presented with two program entities, the blockchain contract (*EV BC contract*) and car's infotainment system. The former is responsible for BC communication and the later provides the user interface for the driver. When some form of interference of driver with the BC communication is needed, the infotainment system and *EV BC contract* communicate via Ethereum BC API. The simplified flow that we are presenting could be implemented as an Ethereum based mobile application. But in hypothetical scenario we have foreseen that the solution is supported by the car manufacturer and is thus fully integrated in the car. Such implementation approach would be the basis for further advancements in the selection algorithm, which would consider for example the current traffic situation and car battery status, suggest the most appropriate driving profile for the selected charging bid, extend the automated charging selection with automated payment, etc.

Figure 2: UML activity diagram of proposed selection procedure

B. System model and operation

Figure 2 presents the UML activity diagram of proposed selection procedure. It is initiated by the *driver*, who selects the route in the vehicle infotainment system (*ITS*). The *ITS* independently of the negotiation stores and updates a directory of CSs (e.g. from a cloud based directory service as presented in Figure 1). Based on the route and their location it creates a list the potential EV CS and passes their Ethereum addresses to *EV BC contract.* The later sends bid request to CSs smart contracts (*EVCS BC contract*) and in this way organizes a kind of a "negative" blind auction. In the example, two possible EVCS are foreseen – EVSC1 and EVCS2. In the bid request bidding duration is set. A bid includes key/value pair that specifies the offered price. When bidding duration expires, the *EV BC contract* selects the best bid and obtains a confirmation by the driver. If this is positive the *EV BC contract* refunds all the non-selected bidders and confirms the selected one.

C. Smart contracts

For the implementation of the proposed solution two smart contracts are required: *EV BC contract* for the vehicle and *EVCS BC contract* for the car. Each of them would implement functions for the activities depicted in Figure 2. In Figure 3 there is a Solidity pseudo declaration for *EVCS BC contract* presented.

```
contract EVCS {

 uint public latitude;
 uint public longitude;

 function eVCS () {
 // initialization of EV charging station
 latitude = 46000000;
 longitude = 15000000;
 }

 function bidRequestReceive () {
 // receives a request for a bid
 }

 function bidSend () {
 // sends a bid to ev
 }
}
```

Figure 3: EVCS smart contract pseudo declaration

IV. CONCLUSION

The elaboration of our idea - even if at the moment presenting just a simplified and partial step toward application of blockchain technologies beyond the payment process in EV charging – indicated us many further interesting research and development directions. In the first step we plan to implement elaborated versions of BC smart contracts and set-up a proof-of-concept system with user interfaces for EV and CS simulation. The next step could be the extension of the model to elaborate the CS reservation process, including penalties if the promised service is not delivered or consumed, and to include the payment process. The selection mechanism is currently reduced to bidding for the lowest price per energy unit. The price however is not the only factor (and often even not the most relevant) influencing charging decisions. Additional parameters could serve as input, e.g. traffic conditions, current battery status, expected charging intensity, anticipated breaks during the drive or additional incentives by the provider to manage more efficiently current bookings of EVCSs. And finally, with closer integration with smart grid, additional mutual benefits for consumer and provider could be achieved. If considering not only charging during the ride, but also at the time where the vehicle is not needed, the EV could serve as temporary energy source/storage for SG. Such a cooperation between EV and SG could lead to new tariff models and reduced prices for EV charging.

ACKNOWLEDGEMENT

The authors wish to acknowledge the support of the research program "Algorithms and Optimization Procedures in Telecommunications", financed by the Slovenian Research Agency.

REFERENCES

[1] M. Swan, "Connected Car: Quantified Self becomes Quantified Car," Journal of Sensor and Actuator Networks, vol. 4, no. 1, pp. 2–29, Feb. 2015.

[2] Z. Yang, "2015 Global electric vehicle trends: Which markets are up (the most)," International Council on Clean Transportation, 18-May-2016. [Online]. Available: http://www.theicct.org/blogs/staff/2015-global-electric-vehicle-trends. [Accessed: 01-Aug-2016].

[3] P. Cazzola and M. Gorner, "Global EV Outlook 2016," International Energy Agency (IEA), Paris, France, May 2016.

[4] V. Pureswaran and P. Brody, "Device democracy- Saving the future of the Internet of Things," IBM, Jul. 2015.

[5] G. Zyskind, O. Nathan, and A. 'Sandy' Pentland, "Enigma: Decentralized Computation Platform with Guaranteed Privacy," MIT Internet Trust Consortium, Jun. 2015.

[6] "Protocol documentation - Bitcoin Wiki." [Online]. Available: https://en.bitcoin.it/wiki/Protocol_documentation. [Accessed: 02-Aug-2016].

[7] V. Trón, "Ethereum Specification," 23-Jul-2015. [Online]. Available: https://github.com/ethereum/go-ethereum/wiki/Ethereum-Specification. [Accessed: 02-Aug-2016].

[8] "Bitcoin - Open source P2P money." [Online]. Available: https://bitcoin.org/en/. [Accessed: 02-Aug-2016].

[9] "White Paper · ethereum," 13-Apr-2016. [Online]. Available: https://github.com/ethereum/wiki/wiki/White-Paper. [Accessed: 02-Aug-2016].

[10] V. Buterin, "SchellingCoin: A Minimal-Trust Universal Data Feed," Ethereum Blog, 28-Mar-2014.

[11] "Solidity — Solidity 0.2.0 documentation." [Online]. Available: http://solidity.readthedocs.io/en/latest/index.html. [Accessed: 31-Aug-2016].

[12] "Solidity realtime compiler and runtime." [Online]. Available: https://ethereum.github.io/browser-solidity/#version=0.3.6. [Accessed: 31-Aug-2016].

[13] C. Greer, et al, "NIST Framework and Roadmap for Smart Grid Interoperability Standards, Release 3.0," National Institute of Standards and Technology, NIST SP 1108r3, Oct. 2014.

[14] "TransActive Grid | Secure, transactional control of utility systems." [Online]. Available: http://transactivegrid.net/. [Accessed: 31-Aug-2016].

[15] J.-P. Buntinx, "Bankymoon to Launch Bitcoin-funded Smart Energy Meter," 10-Mar-2016. [Online]. Available: https://news.bitcoin.com/bankymoon-bitcoin-smart-meter/. [Accessed: 31-Aug-2016].

[16] S. Higgins, "Why a German Power Company is Using Ethereum to Test Blockchain Car Charging," CoinDesk, 07-Mar-2016. [Online]. Available: http://www.coindesk.com/german-utility-company-turns-to-blockchain-amid-shifting-energy-landscape/. [Accessed: 31-Aug-2016].

[17] I. Allison, "RWE and Slock.it – Electric cars using Ethereum wallets can recharge by induction at traffic lights," International Business Times UK, 22-Feb-2016. [Online]. Available: http://www.ibtimes.co.uk/rwe-slock-it-electric-cars-using-ethereum-wallets-can-recharge-by-induction-traffic-lights-1545220. [Accessed: 31-Aug-2016].

[18] "PlugShare." [Online]. Available: https://www.plugshare.com/. [Accessed: 31-Aug-2016].