A Blockchain Approach towards Cargo Sharing in Last Mile Logistics

Johannes Kretzschmar

Institute for Mathematics and
Computer Science
Friedrich Schiller University
Jena, Germany
Email: `johannes.kretzschmar@uni-jena.de`

Felix Eckardt

Institute for Mathematics and
Computer Science
Friedrich Schiller University
Jena, Germany
Email: `felix.eckardt@uni-jena.de`

***Abstract*—Sharing platforms for freight orders are widely used in logistics today. Through the cross-company sharing, resources can be saved ecologically and economically in the long term. In the field of urban logistics however, such platforms are difficult to implement because the operational scope and profit margin are much smaller. This paper presents a method to implement cargo sharing in the Last Mile context using a peer-to-peer (P2P) network solution based on blockchains and smart contracts. Blockchains allow a secure, verified and consensus-based exchange of information while smart contracts ensure the organization of matchmaking, reliable contracting and order execution. The tight linkage of both technologies to cryptocurrencies and existing platforms also ensures the possibility of easy financial balancing and protecting general conditions.**

Keywords–Blockchain; Last Mile; Logistics; Multiuse.

I. INTRODUCTION

The field of logistics is particularly affected by changes over the last years, especially the so-called Last Mile. This term refers to the last step of long supply and delivery chains, which are characterized by short-distance transport processes from a central starting to multiple endpoints or vice versa. There are some critical aspects that require a rethinking and redesign of these kinds of logistics:
The ongoing demographic change and shifting of medial reception from print to online disrupt the business model of media distribution. Subscriptions are a mainstay of the newspaper industry and the omission of these over the last decade critically affects the economic efficiency of distribution tours. While the delivery of newspapers is becoming increasingly unprofitable, the demand for deliveries and services on the last mile is increasing. The displacement towards online shopping and service platforms requires a more extensive infrastructure of Courier, Express, and Package (CEP) transport providers for everyday applications like food retailers or other local suppliers like pharmacies. Finally, there is a strong movement towards environmental protection, especially pollutant and noise emissions. Last mile processes are mostly embedded in urban areas, which are more and more protected by constrained delivery areas or times as well as driving restrictions or even no driving areas for specific vehicles. There is a foreseeable development towards using vehicles without combustion engines. Alternatives hereby are drives from renewable energies like electro mobile or hybrid-driven cars or even new kinds like pedelecs, which have to be evaluated and taken into account for future tour planning.

The research project Smart Distribution Logistik (SDL)[1], on which this paper is based, investigates how newspaper publishers and distributors can address the issues mentioned above. The objective is to determine under which conditions and with which strategies delivery processes can be established long-term cost-efficiently in the future. There are several entangled approaches like open up new business fields, adapting tours, re-slicing delivery areas, reconsidering the vehicle fleet or implementing new logistics concepts like hubs. All these in-house approaches are of course limited. Studies like [2] have shown that the future of last mile logistics lies in cross-company optimization. The goal is to utilize resources, such as electrically driven vehicles as comprehensively as possible to increase cost efficiency.

In the next Section II, fundamental terms and currently used technologies will be introduced to illustrate the context of this work. In the following Section III , a communication platform will be presented, which matches supplies and demands in services and resources of companies to find possible synergies. In the course of discussing the functionality, we present a basic data model for making inquiries in Section III-A and discuss the implementation into a smart contract operating on a blockchain in Section III-B. Further, we present in Section III-C an architecture concept of how to combine a blockchain client with logistic data sources. We conclude this paper in Section IV with discussing several key factors, which such a platform has to fulfill in the context of Last Mile logistics and show how these can be achieved with blockchain technology. While working on the demonstrator, several modeling decisions were made, which are a basis for future work presented in the final Section V.

II. STATE OF THE ART IN CROSS-COMPANY LOGISTICS

With the advent and development of digital technologies, data and networks, a whole range of new products and services have emerged. The systematic cross-linking of hard- and software revolutionized entire supply chains based on monetizable data and data processing. This led to a disruption of whole market segments under the concept of platform economy. These platforms are characterized by their functionality as a central digital marketplace that matches supply and demand and thus brings together various stakeholders [3]. Contrary to the participants of the traditional market, platform operators do not have to bring any actual resources in the segment but only offer the service of matchmaking.

A. Logistics Platform Economy

Logistics has also proven to be a successful application for sharing platforms. The term 4th Party Logistics (4PL) was coined at an early stage in this field to describe the coordination and mediation of logistics service providers and infrastructure. The involved stakeholders are classified into the lower layers 3PL for partners with assets in supply chain management, 2PL for service providers in transport and storage handling and 1PL for manufacturing companies without logistic resources. The different categories may overlap and play various roles, as described in detail in [4]. Of course, the concept of 4PL is closely linked to platform economy, as early direct technological implementations of a marketplace for logistics services, such as [5], show.

Today, there is a large number of commercial solutions in this sector, which are widely used. These arise either from existing logistic service providers who expand their business model in the field of logistics IT to include a corresponding freight exchange or marketplace functionality. These providers like *TimoCom*, *Trans.eu* or *Teleroute* mostly originate from long-haul transports and offer a freight exchange platform specialized in this field. Despite, today there also exist a lot of start-ups in this area, offering services, which target a more private sector like *Shiply*, *uShip* or *Saloodo*.

As described in [3] sharing economy influences traditional markets as well as logistics in particular two ways. On one hand, the market opens to private individuals offering resources or services. On the other, sharing platforms quickly tend to play significant and market-determining roles in their segment like *Airbnb* in accommodation or *Uber* in passenger transportation. The study [6] estimates, that overall sharing revenue will potentially increase from 15 billion in 2015 to 335 billion by 2025. The drawback of sharing platforms in commercial use are the fees of up to 30% of the service price. Because of the high inefficiency due to empty trips, potential large quantities and relatively short detours in traditional long-distance logistics there is a margin for platform prices. Last mile instead is characterized by low quantities, limited cargo space, no dedicated return trips, short distances and complex tours with tight time constraints. These aspects result in much more complex matchmaking but lower gain, which additionally inherits the problem of micro-payments, in a scenario with even now precarious costs. Therefore, we propose an alternative by extending the platform functionality from a central operator to a distributed approach, controlled directly by the service provider and customers.

B. Blockchains in Logistics Applications

One method to distribute applications is distributed ledger technology like blockchains [7]. A blockchain is a linked list of data records that is continuously expanded. Cryptographic procedures are used to ensure that the concatenation and content of the blocks are permanently and immutably fixed. A certain consensus procedure allows several parties to establish and use a uniform database via a peer-to-peer (P2P) network without a proprietary central operator by means of blockchains. In addition to independence and the associated cost savings, blockchains offer further advantages. The distribution of the data can bypass technological bottlenecks of a central network structure and lead to higher reliability. Even if the blockchain itself is publicly accessible, suitable signature procedures can be used to ensure that only certain partners can access data. Due to the missing central organization of the partners, blockchains can be used anonymously by identification with a public key.

Since the publication of blockchains in [7] as part of a technology to implement the cryptocurrency Bitcoin, this technology was applied in various areas. A specific application, which has proved to be very promising is logistics as shown in the study [8] by Hackius and Petersen. Although here, blockchains are intended at a high level to ensure information management between various partners in long and complex supply chains. The benefits of the application are mainly seen in the ease of paperwork processing by using a consistently accessible data structure, identifying counterfeit products by verification or operating internet of things devices. In this paper however, we intend to use blockchains in a much more specific and operative scenario. We will show, that this technology can be the backbone of a P2P sharing network to overcome the problems of last mile logistics sharing mentioned in Section II-A.

C. Smart Contracts and Blockchain

In many cases, such as logistics, it has been shown that the concept of atomic transactions in the original blockchain implementation as introduced in [7] are not sufficient. Often business operations are based on prolonged interactions, which are controlled in complex processes and structures. For this purpose, some blockchain implementations like Ethereum [9] were extended by so-called smart contracts. Smart contracts are Turing-complete programs, which can be instantiated and used by blockchain users. The dependencies and characteristics of an application are modeled into methods, which operate on the blockchain and interact with network peers. The contracts are translated into a bytecode language and executed by an Ethereum Virtual Machine (EVM) on all nodes of the network. In this way, the integrity of the database is permanently ensured, as in the original blockchain approach. The execution of smart contracts costs a cryptoamount, which is clearly defined for each function and loosely coupled to the cryptocurrency ETH of underlying the Ethereum blockchain. This ensures efficient modeling of smart contracts and provides an incentive system for evaluation and block propagation. The in Ethereum common programming language Solidity allows a multitude of complex distributed applications (Dapps), such as independent organizational structures for NGOs, infrastructure for independent voting systems or generic platform economies [10]. In [11], Bogner et al describe an implementation of a sharing platform for rental services for example.

III. DISTRIBUTED MULTI-USE IN LAST MILE LOGISTICS

The goal of this work is the implementation of a platform functionality in logistics via a combination of an adapted blockchain and smart contract technology as advancement to a sharing platform as illustrated in Figure 1. The focus is on the avoidance of a proprietary provider with corresponding costs due to infrastructure, service provisions and eventually the danger of reliance on a market monopoly position. Besides, there should be full control over data publishing, storage and a transparent process model through open smart contracts. Compared to a platform solution, the P2P approach should avoid the risk of infrastructure failure with increased performance and scalability.

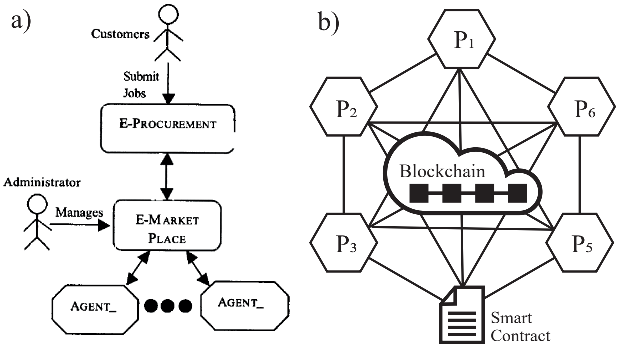

Figure 1. a) Architecture of a 4PL Sharing Platform from [5] compared to b) the distributed ledger P2P Network presented in this paper

A. Request and Offer Representation

The functionality of the network concentrates mainly on the intermediation of requests and orders between various partners. In this case, it covers all fundamental aspects of cargo sharing in a multi-use scenario. The roles of the partners can be distributed across all PL levels, as well as professional companies or private contractors. As usual in a last mile scenario, an order A consists of finite suborders $a_i \in A$. All a_i contain a tuple $< s_i, e_i >$ with the required departure and endpoint. Depending on whether A is a delivery or collecting tour, all departure or endpoints are usually identical and can be determined with s_A or e_A and the set of endpoints or departure points with E_A or S_A. A corresponding example scenario of three logistic partners and a request is illustrated in Figure 2.

Without loss of generality, we focus on delivery tours with a specific departure point s_A and set of endpoints E_A. A logistic partner $L_i \in L$ creates offers by integrating s_A and various subsets of E_A into their existing tours (or creating new ones) and value the costs with a function $o_i(\mathfrak{P}(E_A)) \mapsto \mathbb{R}$ over the power set of endpoint combinations. The matchmaking now finds a minimum set $\bigcup_i P_i = E$ of disjunctive subsets of E, so the sum of the cost of this partition $\sum_i o_i(E_i)$ becomes minimal.

Compared to real-world logistics requirements like weight, size, or temporal constraints, this basic request and offer model is not sufficient of course. However, it is a start for implementing a data structure in the smart contract for representing the fundamentals of interaction with and between blockchain peers.

B. Smart Contract Implementation

The corresponding smart contract and Ethereum client for a sharing functionality based on the model described above was implemented in Solidity 0.4.21 in a test environment provided by the Truffle Framework[12]. Truffle is a comprehensive toolbox for developing, testing and deploying smart contracts on a local Ethereum blockchain.

In the following, a minimal example will be explained to discuss various modeling decisions and correlations. First, a data model is declared in order to store and manage requests and offers on the blockchain via the contract. In addition to a generic data field, the status of a request, a mapping and an iterable array for offers are defined. The structure `Request` as defined in Figure 3 can now be used in the `placeRequest` method for initializing new requests as shown in Figure 4.

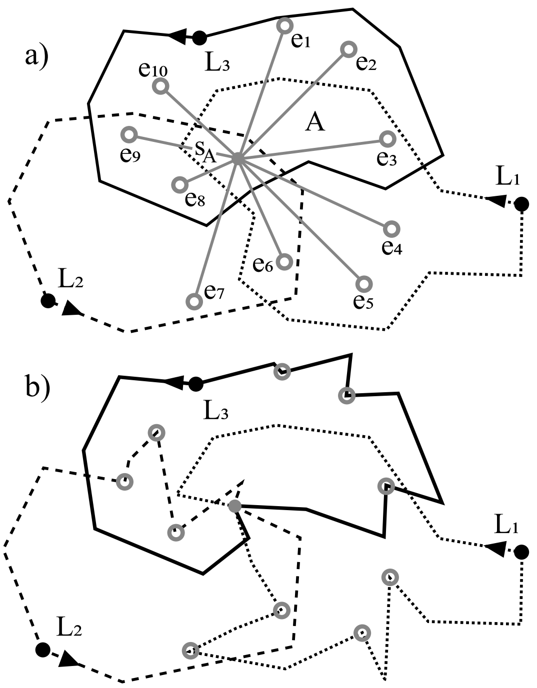

Figure 2. illustrated example of a) a request A in a logistics scenario and b) solution for tour integration

```
struct Request {
     string data;
     bool closed;
     bool completed;
     AcceptedOffers[] accepted;
     mapping(address=>string[]) offers;
     address[] offerers; }
```

Figure 3. Definition of a Data Structure for Requests

Placing a request involves the generation of a new `Request` instance, storing the data and calling a corresponding event linking to the request for notifying the P2P peers and potential contractors. The creation of offers works quite similar as shown in Figure 5. First, a `require` command checks the precondition of a valid open `Request`. Then, the new offer is attached to the `Request`.

The last example in Figure 6 illustrates the method for accepting offers. Preconditions of a valid `RequestID` for an

```
function placeRequest(string data) public
     returns (uint id) {
  uint RequestID = Requests[msg.sender].length++;
  Requests[msg.sender][orderID].data = data;
  emit NewRequestPlaced(msg.sender, RequestID);
  return RequestID; }
```

Figure 4. Method for Placing Requests

```
function placeOffer(address Requester,
    uint requestID, string data)
    public returns (uint id) {
  require(Requests[Requester].length > requestID);
  Request storage o = Requests[requester][requestID];
  require(o.closed == false);
  if(o.offers[msg.sender].length == 0) {
    o.offerers.push(msg.sender);
    }
  o.offers[msg.sender].push(data);
  return o.offers[msg.sender].length - 1; }
```

Figure 5. Method for Placing Offers

```
function acceptOffer(uint requestID,
    address offerer, uint offerID) public {
  require(Requests[msg.sender].length > requestID);
  Order storage o = orders[msg.sender][orderID];
  require(o.closed == true && o.completed == false);
  require(o.offers[offerer].length > offerID);
  uint index = o.accepted.length++;
  o.accepted[index].offerer = offerer;
  o.accepted[index].id = offerID; }
```

Figure 6. Method for Accepting Offers

open request as well as a valid `offerID` are checked. After that, the offer is added to the accepted offers of this request. These simple examples do not include a registration and signature check at the beginning to verify whether a user actually has access to the smart contract or a method. Only the originator is allowed to modify or withdraw requests or offers.

In the current state of the demonstrator, all data is stored in the blockchain for validation. However, it has been shown that this is not optimal in this application scenario. Due to a large number of possible offers, which depends on the number of elements in a request and the number of service providers, the storage effort increases exponentially. This leads to an inefficient execution of the smart contract, because storage operations on the blockchain are cost-intensive, and results into intense memory needs for storing the blockchain locally as well as higher network traffic for the propagation of new blocks. An implementation is currently being tested in which only requests are written to the blockchain and the offer phase is implemented via Whisper channels. The P2P users communicate directly with each other via the Ethereum specific communication protocol Whisper. Each client manages given and received offers independently locally. After the expiration of the request deadline the accepted offers calculated my the matching algorithm are stored on the blockchain and the contractors are notified via a corresponding event trigger or Whisper message. This contradicts the claim of a blockchain solution that all aspects of a contract are reproducible and verifiable stored but avoids the problem of storage effort and load.

C. The P2P Client Integration

In order to make the blockchain application as accessible as possible, an initial client implementation was created in connection with an Enterprise Resource Planning (ERP) system and user interface. A first approach primarily serves as a technological proof-of-concept. In the long term, the current status and history of the blockchain, as well as smart contracts, should be visualized user-friendly and operable. It also should be possible to define event triggers, enable automated requests deployments, evaluations and submissions of offers.

Based on the given Ethereum client implementation, an interface to a logistics ERP system was created, which contains a number of existing tours, orders and fleet information. This extended client is multifunctional with regard to a requester, as well as a offerer. The basic architecture is shown in Figure 7. A set of orders is converted from ERP data into a request specific format by the RequestHandler. The RequestHandler then publishes and manages the request on the blockchain via the smart contract methods as described above. Depending on the implementation, incoming offers are buffered by the RequestHandler locally or written on the blockchain directly by the contractor and evaluated by the RequestHandler at the end of the expiration time. The RequestHandler selects a set of offers based on predefined criteria in the business rule model and closes the request using the smart contract. The now implemented rules only choose offers according to minimize costs. Eventually, the smart contract triggers an event that notifies all bidders of the status of their bids.

In addition, the client reacts to events from the blockchain, such as newly generated external requests by the OfferHandler. This handler evaluates a request by incorporating each combination of subsets into existing tours from the ERP system as described in Section III-A. If there are possible tour combinations, the handler generates offers by evaluating the tour changes. The costs are calculated depending on the increased effort caused by the tour change and profit intentions. The effort is calculated using a Total Cost of Ownership (TCO) model that was developed in SDL specifically for logistics scenarios and whose integration is described in [13]. The profit intentions are stored in the business rule base along with other factors, such as under which circumstances an offer is to be submitted. The OfferHandler also evaluates blockchain events in case an offer is accepted to integrate the changes into the ERP system.

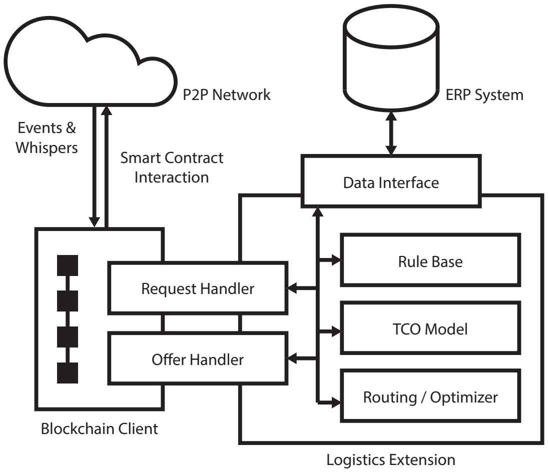

Figure 7. Components of the extended Blockchain Client

IV. CONCLUSION

In this work, we were able to present in a proof-of-concept that the marketplace functionality of a sharing platform for logistic goods can be implemented in the last mile context using blockchains. By doing so, service providers and clients are not dependent on a cost-intensive central proprietary platform but can conduct business via an autonomous consensual P2P network. The costs are therefore limited exclusively to the computing effort involved in expanding and storing the blockchain. The P2P approach ensures an easy ad hoc access for sharing resources and the distributed application implies high reliability and scalability. Eventually, the work on the demonstrator posed a few interesting modeling decisions and issues in multiple fields, which will be tackled in the future work within the SDL project.

V. FUTURE WORK

In addition to the development of a basic demonstrator, this work laid the foundation for further issues in research and development. Three basic problem areas were identified, which have to be covered for a successful application of a blockchain-based logistics sharing platform:

First, various aspects of the application domain must be clarified. The current model and methods are based on very simple and rudimentary assumptions. For a tangible application, a general uniform data model must be developed that covers all possible aspects of a last mile delivery, such as time, size or weight restrictions or additional requirements, such as cooling or tracking of transport. There is also the question of heuristics about the submission of offers, calculation of the profit share and which offers are accepted. Is a requester really looking for the cheapest solution or the smallest possible number of service providers to simplify scheduling and organization at the ramp? These issues are not blockchain specific but affect all sharing platforms in this application field. So it is foreseeable that under the research topic of 4PL there has to be an open standard covering data exchange aspects.

The question of legal regulations must also be addressed. The question of whether and how contracts can be (semi-)automatically concluded and under which conditions they are binding is currently still a research topic and far off from everyday use.

Eventually, there is the question of how and where the blockchain and thus the smart contract is deployed. A private blockchain with restricted access is conceivable and allows full control over deployment parameters and functionality but implies an organizational structure for registration and authentication. The implementation of such a central infrastructure would be opposed to the distributed concept of blockchains. An openly accessible but application-specific blockchain, on the other hand, requires a critical number of participants due to the danger of consensus attacks, which can compromise a blockchain. The third and most promising possibility is the deployment into the existing Ethereum blockchain. The main advantage here is the opportunity of using the established cryptocurrency ETH. This currency can be used directly in the logistics application to reimburse services or enable defining penalties in the contract. A disadvantage, however, is that there would be no control of aspects like the consensus algorithm for example and a heavy dependency on the actual ETH currency rate.

ACKNOWLEDGMENT

The project Smart Distribution Logistik (SDL), on which this paper is based, was funded by the Federal Ministry of Economics and Energy under the promotional reference number 01ME17001C. The responsibility for the content of this publication lies with the author.

REFERENCES

[1] "Smart distribution logistik website," http://www.sdl-projekt.de, accessed: 2019-05-10.

[2] A. Juan, C. Mndez, J. Faulin, J. Armas, and S. Grasman, "Electric vehicles in logistics and transportation: A survey on emerging environmental, strategic, and operational challenges," Energies, vol. 9, 01 2016, p. 86.

[3] K. Stanoevska-Slabeva, V. Lenz-Kesekamp, and V. Suter, "Platforms and the sharing economy: An analysis eu h2020 research project ps2share: Participation, privacy, and power in the sharing economy, 2017," SSRN Electronic Journal, Nov 2017.

[4] L. Saglietto, "Towards a classification of fourth party logistics (4pl)," in Universal Journal of Industrial and Business Management, vol. 1, Jan 2013, pp. 104 – 116.

[5] H. C. Lau and Y. G. Goh, "An intelligent brokering system to support multi-agent web-based 4/sup th/-party logistics," in 14th IEEE International Conference on Tools with Artificial Intelligence, 2002. (ICTAI 2002). Proceedings., Nov 2002, pp. 154–161.

[6] PwC, "The sharing economy–sizing the revenue opportunity," 2015.

[7] S. Nakamoto, "Bitcoin: A peer-to-peer electronic cash system," 2009, accessed: 2019-05-10. [Online]. Available: http://www.bitcoin.org/bitcoin.pdf

[8] N. Hackius and M. Petersen, "Blockchain in logistics and supply chain : trick or treat?" in Proceedings of the Hamburg International Conference of Logistics (HICL), 2017, pp. 3–18.

[9] G. Wood, "Ethereum: A secure decentralised generalised transaction ledger," Ethereum project yellow paper, vol. 151, 2014, pp. 1–32, accessed: 2019-05-10.

[10] M. Mainelli and M. Smith, "Sharing ledgers for sharing economies: an exploration of mutual distributed ledgers (aka blockchain technology)," Journal of Financial Perspectives, vol. 3, no. 3, 2015.

[11] A. Bogner, M. Chanson, and A. Meeuw, "A decentralised sharing app running a smart contract on the ethereum blockchain," 11 2016, pp. 177–178.

[12] "Truffle suite framework website," https://truffleframework.com/, accessed: 2019-05-10.

[13] J. Kretzschmar, M. Johlke, and W. Rossak, "A tco analysis tool based on constraint systems for city logistics," in 7th International Conference on Advances in Vehicular Systems, Technologies and Applications, 2018 (VEHICULAR 2018). Proceedings., 2018, pp. 37–38.