Available online at www.sciencedirect.com

ScienceDirect

Procedia CIRP 79 (2019) 456–460

www.elsevier.com/locate/procedia

12th CIRP Conference on Intelligent Computation in Manufacturing Engineering, 18-20 July 2018, Gulf of Naples, Italy

# Framework for enabling order management process in a decentralized production network based on the blockchain-technology

Wjatscheslav Baumung[a,*], Vladislav Fomin[b,c]

[a]*Reutlingen University, Alteburgstraße 150, 72762 Reutlingen, Germany*
[b]*Vytautas Magnus University, Donelaičio g. 58, 44248 Kaunas, Lithuania*
[c]*Vilnius University, Muitines g. 8, 44280 Kaunas, Lithuania*

* Corresponding author. Tel.: +49-7121-271-4057; E-mail address: wjatscheslav.baumung@reutlingen-university.de

**Abstract**

The promise of immutable documents to make it easier and less expensive for consumers and producers to collaborate in a verifiable way would represent an enormous progress, especially as companies strive for establish service contracts which are based on the flow of many small transactions using machine-to-machine communication. The blockchain technology logs these data, verifies the authenticity and make them available for service offers. This work deal with an architecture enabling to setup order processing between consumers and producers using blockchain. In this way, the technical feasibility is shown and the special characteristics of blockchain production networks will be discussed.


Peer-review under responsibility of the scientific committee of the 12th CIRP Conference on Intelligent Computation in Manufacturing Engineering.

*Keywords:* Production network; Blockchain; Smart contract; Secure data exchange; Additive manufacturing

## 1. Introduction

The Blockchain is a new technology which enables transactions to be processed in a decentralized, manipulation-secure and transparent form for all members of a network. The blockchain relies on decentralization by storing a series of data records (blocks) via an individual concatenation (hash values), always on its predecessor or successor data record [1]. These so-called "hash" sentences form the connection between the blocks, in other words the chain. This creates a decentralized database with a constantly growing list of transaction data records. Especially in the financial sector, the blockchain is currently seen as a disruptive key technology, whose application possibilities, however, go far beyond this sector [2]. Besides the pure storage of transactions, as with the Bitcoin network, some other platforms such as Ethereum provide the storage and execution of so-called Smart Contracts in the blockchain. Smart contracts are computer programs that map contracts and check their compliance. The use of Smart Contracts opens new perspectives for manufacturing companies, as critical data such as design drawings or orders can be sent securely across company boundaries. In the future, the security of product and order data, which is increasingly perceived as strategic resources of companies, will take priority. Blockchain technology is able to protect complete production chains against unauthorized access and thus increases the possibility of network cooperation [3]. The sharing of production data for efficient utilization of capacities in the network with the help of blockchain technology is shown below using the example of multi-dimensional production facilities[1]. Companies face several challenges in the acquisition of innovative technologies such as additive manufacturing technology. These include high initial investments, the development of know-how, and the risk of unused capacities at low capacity utilization [4]. In some

[1] Multidimensional manufacturing processes describe flexible technologies that can influence the productivity of capacity utilization due to their arrangement in 2- or 3-dimensional space. For example, plasma cutting belongs to 2-dimensional manufacturing and additive manufacturing processes to 3-dimensional manufacturing

2212-8271 © 2019 The Authors. Published by Elsevier B.V.
Peer-review under responsibility of the scientific committee of the 12th CIRP Conference on Intelligent Computation in Manufacturing Engineering.
10.1016/j.procir.2019.02.121

cases, high market prices from suppliers prevent access to the integration of innovative production technologies into the company's own added value [5]. Thereby the utilization of the production facilities is analyzed and optimized through planning and control systems exclusively within a company boundary. In contrast, the cross-company and access-protected data exchange of product and order data enables new cross-sector potential benefits to be generated. The outsourcing of production orders is usually associated with considerable organizational effort and is therefore not transparent with limited flexibility. Another way of meeting the required delivery dates in a constantly growing order situation is to increase the company's own production capacity. However, there is a risk that production capacities will be used inefficiently [6].

The architecture presented in this paper supports manufacturing companies and customers by connecting the two parties without intermediaries on the one hand and enabling an economic optimization of production capacities via an open blockchain-based production network with secure product and order data transfers on the other.

## 2. Architecture for a blockchain-based production network

The figure 1 describes the simplified overview of an architecture for building a blockchain-based production network for order processing using additive manufacturing (AM) and subtractive manufacturing (SM). The aim is to establish a structure that enables mutual interaction between the producer and the customer. The data required for this (Fig. 1 - 1.) is stored in so-called Smart Contracts (SC) in a manipulation-proof and transparent format (Fig. 1 - 2.). This data contains the address data, order details on the customer side and the address data or capacity descriptions on the producer side. A software takes over the combination of supply and demand in order to save the data for the order as a new SC (Fig. 3 - 3.). A suitable production capacity can be determined on the customer side and customer orders can be found on the producer side to increase the utilization of a production capacity. In the following, the architecture is explained in detail from the perspective of the customer and the producer.

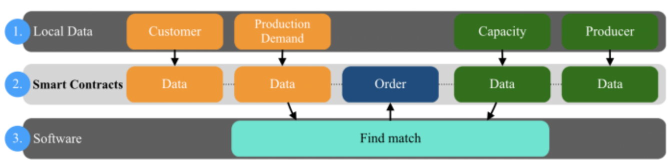


Fig. 1. Simplified overview of the blockchain-based production network.

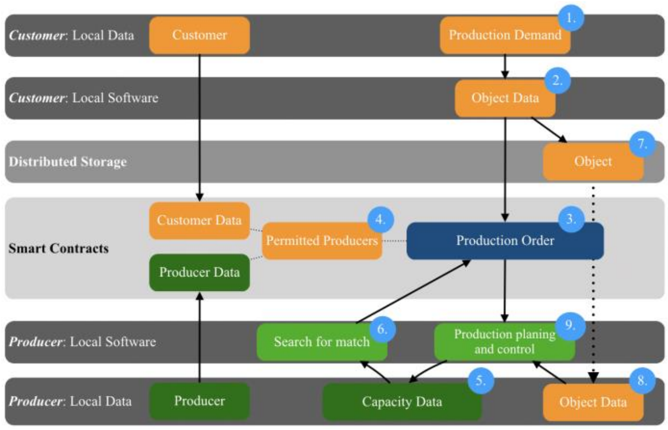


Fig. 2. Order generation according to customer demand.

### 2.1. Order creation based on the customer's demand

Figure 2 explains how an order is generated based on the customer's demand to produce goods (Fig. 2 - 1.), which is referred to as the production object in the following. To make the production object available to producers in an SC, it must first be transferred to the form of a SC. This requires a software that can process the data and then store it as SC in the blockchain (Fig. 2 - 2.). The software takes over the recording of public and private data. The public data provides the information that the producer needs to decide whether he wants to accept the order. Therefore, the public data includes the dimensions of the building space which can be taken from the production object. In the most basic case, these data is an outline with the maximum values, for SM in X and Y direction and for AM additionally in Z direction. The outline can also be formed from several points, which increases the probability of finding a suitable capacity at the producer. The private data represent the production object itself, ensuring that it can be viewed only by the producer who receives the order. The public data is saved together with a placeholder for the path to the production object as the "Production Order" SC (Fig. 2 - 3.). The SC "Permitted Producers" represents the restriction which producers are allowed for the production order (Fig. 2 - 4.). On the producer side, the Planning and Control System (PPC) reports that capacity is not completely used, for example due to a planned production order (Fig. 2 - 5.). This data is transferred into a uniform structure, such as the available material, utilization lot and, in the case of AM, the usage time. Afterwards, the system searches for suitable orders that meet the criteria mentioned (Fig. 2 - 6.). This step can be done manually or by software. In the case of software used, this enables machine-to-machine communication between the customer and the producer. If there is a match between a customer and a producer, the offer is recorded by the producer in the SC "Production Order" (Fig. 2 - 3.). Other producers can continue to send offers to SC "Production Order" as long as the bid price is below the previous quotation price. No further bids can be submitted if the customer indicates in the SC "Production Order" that the tender has ended. The public key of the producer is read from the SC "Producer Data" and used for the encryption of the production object. After encryption, the production object is stored in a decentralized storage and the path to the encrypted production object is stored in the SC "Production Order" (Fig. 2 - 7.). The producer then receives access to the production object via the path (Fig. 2 - 8.) and can decrypt it with his private key. The PPC will then take the production object into account (Fig. 2 - 9.). If the producer still has unused capacity, the process can start again at the producer's "Capacity Data" (Fig. 2 - 5.).

*2.2. Representing an intermediate contractor*

Figure 3 describes a structure in which the customer states that only the producers he has specified in his SC production demand (Fig. 3 - 2.) are allowed to cover his production needs (Fig. 3 - 1.). This can be useful, for example, to ensure quality, since the customer only accepts producers for his orders with whom he previously had personal contact or could convince himself of the quality beforehand. In order not to restrict potential producers too much, the architecture presented (Fig. 2) allows a recursive structure. Thus, the permitted producers (Fig. 3 - 3.) can map the production order as a production requirement (Fig. 3 - 4.) and make it available to their producers (Fig. 3 - 5.). The authorized producers can access the production order and submit a bid for it (Fig. 3 - 6.). If a producer is awarded the contract for the production order via the subcontractor, the production object does not have to be stored again. In the newly added SC (Fig. 3 - 5.) only the reference to the original SC (Fig. 3 - 2.) with the production path is referenced. The advantage here is that partner relationships can be mapped, e.g. a partner abroad takes over the development of production resources via local producers. The transparent and tamper-proof architecture by means of the block chain and the SCs take into account the respective liability of the producers. Thus, the producers (Fig. 3 - 6.) assume liability and quality assurance towards the client (Fig. 3 - 5.), while the client itself (Fig. 3 - 3.) assumes liability and quality assurance towards the original customer.

*2.3. Order processing on the basis of the manufacturer's available capacity*

Figure 4 describes a structure in which the manufacturer makes his manufacturing capacity available to the production network. The focus is shifted to the manufacturer where the capacity is completely unused. Therefore, this structure describes making the capacity available by means of an SC for a period of time instead of finding all the individual production orders for this capacity together.

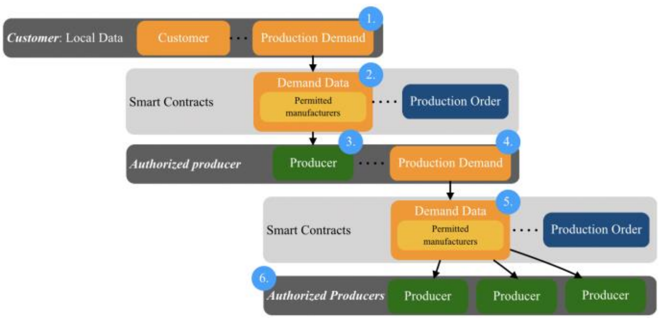


Fig. 3. Recursive structure for the representation of an intermediate contractor.

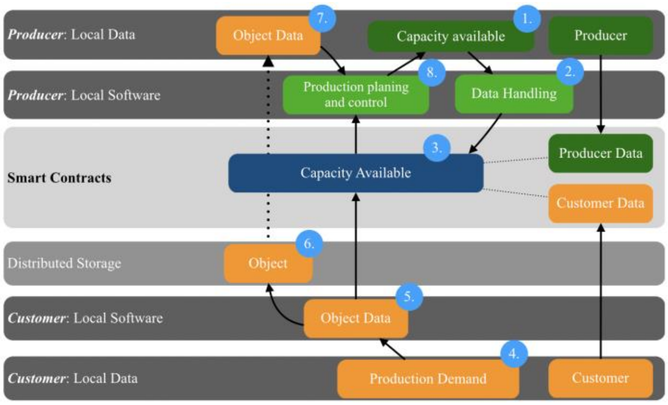


Fig. 4. Structure based on the manufacturer's available production capacities.

The start is triggered by the PPC, which also has an available capacity (Fig. 4 - 1.). The capacity is prepared with software in a uniform form for the SC (Fig. 4 - 2.). As a result, all public data on the production capacity, such as production process, machining area, available materials and service life, are stored in the blockchain in the SC "Capacity Available" (Fig. 4 - 3.).

On the customer's side, triggered by the production demand (Figs. 4 - 4.), all available production capacities can be read out using the SCs "Capacity Available" using software and the most suitable manufacturer can be selected based on the required production criteria (Figs. 4 - 5.). The customer can then place a bid for the capacity (Figs. 4 - 3.). Further bids for the capacity may be submitted as long as they are higher than the previous ones and the producer has not completed the bidding. The customer who has been awarded the contract determines which production objects are to be used for the production capacity and stores them encrypted in a decentralized storage (Figs. 4 - 6.). Encryption is performed using the public key of the producer located in the SC "Producer Data". The generated path to the production objects is then saved in the SC "Capacity Available" for the production order (Figs. 4 - 3.). After decoding, the producer has access to the individual production objects (Figs. 4 - 7.) and can take them into account in his PPC (Figs. 4 - 8.).

## 3. Implementation

For the simulation of the presented architecture, a web portal was developed that represents both the customer side and the producer side. For the Public Blockchain the Ethereum Platform was used, where the SC was written with the language Solidity [7] developed by Ethereum. The local software for the producer and customer was developed exclusively web-based with Javascript to be able to use the web3.js [8] from Ethereum. Web3.js allows interaction with the SCs over an HTTP or IPC connection. Thus the browser plugin Metamask [9] could also be used for authentication and interaction with the SCs in the browser. For simulating the blockchain, Ganache was used to test the functionality of the SCs. Ganache [10] represents a private blockchain for development for Ethereum.

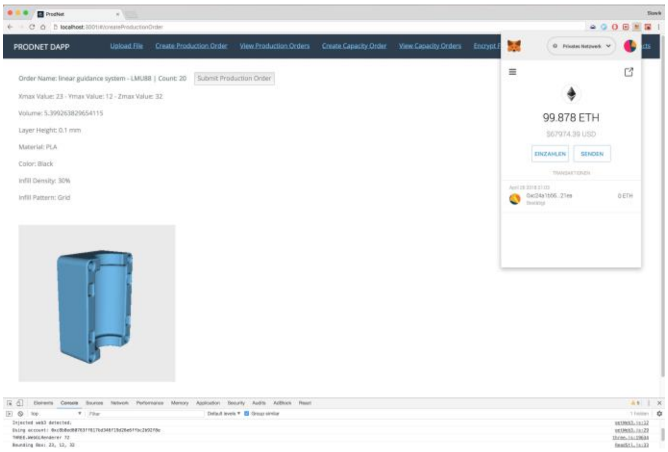


Fig. 5. Development of a web portal to simulate customer and producer views.

The compilation and migration of the Smart Contracts was done with Truffel [11]. Truffel is a development environment and test framework for the Ethereum.

From the customer's point of view, the file upload of the production object is the starting point. After the manufacturing object was uploaded to the web server, the package node-stl [12] for AM and SVG.js [13] for SM were used to calculate the bounding boxes to determine the required installation space. The result is the maximum values for the axes X, Y for SM and additionally Z for AM. In addition, the volume of the production object was calculated for AM. In order to save the data in SC, the customer must first authenticate himself with his account via Metamask. After authentication, the data was then stored together with the production data in the SC production requirement. The customer can use another SC to determine which producers can place bids. The standardized data format OpenPGP was used to encrypt the manufacturing objects [14]. After a producer was found for the job, his public key was read directly from the producer data and the encryption for this offer in the browser with OpenPGP.js [15] was performed. The encrypted manufacturing object was then stored decentrally using IPFS [16], which identifies the files using a hash value. This hash value was then stored in a private variable in the Smart Contract production demand. On the web portal for the producer, the producer alone could read the private variable with the hash value after authentication via Metamask. Using the IPFS hash value, he could then download the encrypted manufacturing object and decrypt the manufacturing object in the web browser by his own private key and save it locally.

## 4. Results

The functionality of the presented architecture could be tested with the implementation. Via the presented web portal, available production capacities were offered by the manufacturer and at the same time production requirements of customers were networked with these production capacities. Although there was only one web portal for customers and producers, it was possible to use Metamask to switch accounts and thus to check the viewpoints for each transaction. Using the Public Blockchain, all producers and all orders with all required production data, such as building space size, manufacturing process and material, could be exchanged and accessed without the need for an intermediary. Contract-dependent data was stored in the Smart Contracts and the problem with the large files for the production objects could be solved with distributed storage using IPFS, since direct storage on the Ethereum platform would lead to high costs. A criticism of this approach was raised by the public blockchain from several points of view. The attitude of the decentralized base requires participants who make their computing power available for the confirmation of transactions. The number of participants influences the security of the network. In order to encourage participants to provide their computing power, they must be rewarded, which leads to fees for the creation and processing of Smart Contracts. From the perspective of companies, the complete transparency of transactions can be a disadvantage and an advantage at the same time, such as the data provided about the customers and producers. The advantage of this approach is that the entire supply chain can remain completely transparent for the customer, but the question arises as to whether this data should really be disclosed transparently from an entrepreneurial point of view.

## 5. Conclusion

The architecture and implementation presented demonstrates how order processing with document exchange via blockchain between customer and producer can be carried out in a manipulation-proof and transparent manner.

It was shown how production facilities of an open production network can support production within the own company. The barriers described in the introduction are thus reduced for companies. At the same time, producers can offer their available production capacities directly on the market. In addition, the individual production plants with their specific characteristics and capacities can be mapped flexibly and as an integrative open production network. Complete transparency in the production network is ensured by order identification with the production data and the data of the addressee, which can be identified at any time using the transaction code. The flow of numerous small-sized transactions could be mapped via the web browser control system. This can be used as the basis for machine-to-machine communication. The novelty of the approach lies in the provision of an open but secure production network using blockchain technology. The example of multidimensional production in particular offers companies a resource-saving opportunity for cooperative cooperation and strengthening individual competitiveness.

## References

[1] Risius M, Spohrer K. A Blockchain Research Framework. Bus Inf Syst Eng 2017;59(6):385–409.

[2] Petersen M, Hackius N, Kersten W. Blockchains für Produktion und Logistik. ZWF 2016;111(10):626–9.

[3] Burgwinkel D (ed.). Blockchain technology: Einführung für Business- und IT Manager. Berlin, Boston: De Gruyter Oldenbourg; 2016.

[4] Gibson I, Rosen D, Stucker B. Additive manufacturing technologies: 3D printing, rapid prototyping and direct digital manufacturing. New York, Heidelberg, Dodrecht, London: Springer; 2015.
[5] Klahn C, Leutenecker B, Meboldt M. Design Strategies for the Process of Additive Manufacturing. Procedia CIRP 2015;36:230–5.
[6] Gebhardt A, Hötter J-S. Additive manufacturing: 3D printing for prototyping and manufacturing. Munich, Hanser Publishers, Cincinnati: Hanser Publications; 2016.
[7] Ethereum. Solidity. https://solidity.readthedocs.io/en/v0.4.23/; 2016.
[8] Ethereum. web3.js. http://web3js.readthedocs.io/en/1.0/index.html; 2016.
[9] Metamask. Metamask. https://github.com/MetaMask/metamask-extension; 2016.
[10] CONSENSYS. Ganache. http://truffleframework.com/ganache; 2018.
[11] CONSENSYS. Truffle. http://truffleframework.com; 2018.
[12] Boyne J, Vazhatharayil AJ, Island Games Studio, lexe11. Node-stl. https://www.npmjs.com/package/node-stl; 2018.
[13] Wout Fierens. SVG.js. http://svgjs.com; 2018.
[14] OpenPGP. OpenPGP. https://www.openpgp.org/about; 2018.
[15] OpenPGP.js. OpenPGP JavaScript Implementation. https://openpgpjs.org; 2018.
[16] Protocol Labs. IPFS. https://ipfs.io; 2018.