Journal of Machine Engineering, 2021, Vol. 21, No. 1, 89–97
ISSN 1895-7595 (Print) ISSN 2391-8071 (Online)

Received: *09 November 2020* / Accepted: *22 December 2020* / *29 March 2021*:

distributed ledger,
smart contract, value networks

Gordon LEMME[1*], Kilian Armin NÖLSCHER[1],
Erhao BEI[2], Christian HERMELING[1],
Steffen IHLENFELDT[2]

SECURE DATA STORAGE AND SERVICE AUTOMATION FOR CYBER PHYSICAL PRODUCTION SYSTEMS THROUGH DISTRIBUTED LEDGER TECHNOLOGIES

In this paper, we use the blockchain technology to design a prototype to secure process data from a 3D-printer. Datastreams are gathered from various sources such as OPC UA servers and autonomous retrofit sensor nodes. This is followed by pre-processing for data reduction, storage in a data model, and the generation of a unique hash value over it. The hash values are stored in a blockchain using appropriate consensus methods, taking into account their temporal origin and production identification number. This also includes the context-related influence of sensor signals on the production process Restrictive access regulations using smart contracts make a partially or fully automated machine tool calibration possible. In this context, we show to realize a process partial or full automation through smart contracts. Physical machine tools and virtual simulations are integrated into the block-chain network to document the stability and performance.

1. INTRODUCTION

The relocation of contracting and supply chains into virtual space, the so-called "cyberspace" is leading to a transformation in manufacturing with the help of cyber-physical production systems (CPPS), which emerged from the digital revolution. The use of CPPS with a high degree of sensor technology and connectivity offers completely new possibilities for digital integration and thus an increased level of automation. Such digital integration enables context-related information to be combined into a virtual image via physical-virtual interfaces. The associated extensive data acquisition enables the recording of the history of a product over a period of time during the manufacturing process [1].

Due to the amount of data as well as the heterogeneous scattering of data sources (e.g., additional sensors for environmental data acquisition and machine control on production systems), new questions arise regarding the secure data exchange and adequate data storage. Since this (meta-) data is particularly in the context of knowledge-based systems coveted

[1] Cyber-Physical Production Systems, Fraunhofer Institute for Machine Tools and Forming Technology IWU, Germany
[2] Chair of Machine Tools Development and Adaptive Controls, TU Dresden, Germany
[*] E-mail: kilian.noelscher@iwu.fraunhofer.de
https://doi.org/10.36897/jme/131917

information in the sense of company-specific, specific process knowledge, it has to be protected against stealing and, concerning later audit measures, against subsequent manipulation [2]. A higher degree of automation is the driving force behind an increase in productivity and flexibility, which is what gave rise to the idea of an independently operating manufacturing system. A system that independently receives and bills orders, initiates maintenance work, checks for legitimacy, and controls processes [3].

Such visions are supported by the successful application of new technologies such as, for example, advances in machine learning, which have taken the topic of "artificial intelligence" to a new level in recent years, and distributed ledgers with their captivating idea of transparency. These promise a high degree of security by design concerning the unchangeability of discrete data packets while at the same time offering a wide variety of settings. A well-known representative of this is the blockchain, which attracted special attention in public with its implementation as a Bitcoin blockchain. Besides, the smart contracts introduced with the Ethereum blockchain have an enormous potential for automation, which has so far mainly been used in the financial world and online trade [4].

In addition to the mentioned innovations in the field of information technology, the emergence of additive manufacturing brought about major changes in product design, prototyping and production [5]. In this group of manufacturing technologies, adherence to defined process parameters is elementary, especially with respect to quality management during production. In this process family, many factors have a direct or indirect influence on the quality of the component. For example, compliance with certain temperature intervals, humidity values of the printing material and the environment, the intensity of the incident light, the use of different batches of printing material and downtime of the system during a printing process [6]. These variables, summarized under the term context information, must be monitored as well as logged and – by the machine operator – compensated [7].

Against the background of manipulation security as well as transparency and automation, a solution is missing to automate the event- and context-dependent control of production systems, which was previously brought in by humans, and to be able to independently log and store the resulting circumstances of the creation of a component. On the way to a Smart Service, the architecture will be designed in the context of this paper to securely store data occurring at CPPS and to be able to trigger event-based control actions via smart contracts.

For this purpose, the relevant technologies are presented in Section 2. In Section 3 the system architecture is presented, and in Section 4 insights into the implementation are given. Section 5 summarizes the results and gives an outlook on further steps.

2. STATE OF THE ART

2.1. CPPS

Cyber-physical production systems, as the evolution of traditional machine tools, offer the possibility of using data-based services due to their connectivity, the wealth of control

information, and communication protocols that extend into the manufacturing network layer [8].

Often such manufacturing systems are kept separate from public networks for security reasons to prevent damage and production downtime due to unauthorized access (hackers). Machine tools are subject to ever-increasing demands for accuracy and repeatability of movement to produce a dimensionally accurate product. For example, the heat generated during the production process and general wear and tear on individual components or assemblies (e.g., joints) of the machine tool during the normal production process influence the accuracy [9].

To meet the requirements of fast data acquisition, solutions from information technology are needed. In our case we use technologies for data buffering through the Kafka framework and distributed data storage in a Mongo database. These technologies are briefly described below. Apache-Kafka allows data to be transferred as a background data stream between different external (data) systems, ensuring efficient (2 million writes per second) and reliable data transfer. Reading, writing, and processing of the data is implemented through Apache-Kafka's four APIs, respectively Kafka Connect API, Kafka Streams API, Kafka Producer API, Kafka Consumer API [10].

MongoDB can be seen as a new kind of relational database. A traditional relational database structures data in relational tables and rows, while MongoDB structures data in collections of relational Json (Javascript Object Notation) documents. This is characterized by properties such as flexibility and speed and can also be created in a distributed manner. These are so-called replica sets, which means that the data sets of one client are stored as copies on other clients in the network, and in the event of a failure of this client, another client takes over the service [11].

Two transmission protocols are used to integrate machine tools and sensor data. These are the machine-to-machine protocol OPC UA (Open Platform Communications Unified Architecture) and the lightweight MQTT protocol. MQTT (Message Queuing Telemetry Transport) is a machine-to-machine protocol based on the publish-subscribe pattern with hierarchically organized channels as a tree structure. In the simplest case, client A publishes its data packet on a channel with which it, or its data packet, can be identified. Client B receives this as soon as it subscribes to the channel or the complete branch the channel is attached to.

A central server, the MQTT-Broker, is responsible for managing the data streams and channels [12]. OPC stands for Open Platform Communications and is one of the most important communication protocols for Industry 4.0 and IIoT. With OPC, access to machines, devices, and other systems in industrial environments is standardized and enables similar and manufacturer-independent data exchange [13]. The UA in OPC-UA stands for “Unified Architecture” and describes the latest specification of the standard.

Based on OPC-UA, the Industry 4.0 communication fits into the reference architecture model for Industry 4.0 (RAMI4.0) of the Industry 4.0 platform [14]. OPC-UA is currently widely used as a popular standard in industrial communication. It offers two basic communication protocols, namely TCP and UDP. TCP provides secure peer-to-peer communication by establishing a channel while UDP provides unreliable broadcast communication [15].

2.2. DISTRIBUTED LEDGERS

Distributed-Ledger-Technologies (DLT) is the collective term for technologies that hold data and transactions in a decentralized manner, i.e., distributed in a network, whereby all changes are adopted by all participants after their verification [16]. An essential feature of distributed ledger technology is the backward linking of individual blocks using a hash value into a mostly linearly linked list (e.g. blockchain). All participants (nodes) of such a distributed ledger network can persist a copy of the ledger locally. Through a consensus mechanism (e.g., Proof of Work, Proof of Stack), the network decides on the acceptance of a block and the updating of the ledger, which can be stored on each node.

Public and private blockchains are both distributed peer-to-peer networks [17], where a public blockchain is characterized by the fact that any interested party can participate without prior authorization. In the context of this paper, a public blockchain is applied to realize the use of services in a tamper-proof way. A private blockchain is only available for a selected group of users so that the network nodes have to be legitimized by a superior instance to participate in the distributed network. Due to the high-security requirements in the production environment, a private distributed ledger network is suitable for internal use within a company, especially at the company level. This allows the use of less computationally intensive, but fast consensus mechanisms to record numerous process parameters in a tamper-proof manner and store them in an auditable manner.

Smart contracts are an essential component of distributed ledger technology for process automation and can be used within a blockchain. [18]. This is a functional extension of the distributed ledger technology, which, according to Szabo [19], forms the basis for a digital, mediation-free equivalent of conventional contracts. However, due to the still young technology, there are no legally sound judgments on this, which is why a smart contract is more likely to be regarded as a source code that works on the “if this-then that” principle. An smart contract within distributed ledger technology thus basically implements a digital, distributed process for the automated handling of if-then conditions. Here, source code with a limited functional scope, which is stored on the blockchain and triggered on an event basis, controls the automation. Smart contracts are stored in the blockchain and, like other objects (e.g. persons, machine tools), are given their address, which can be addressed by each participant in the blockchain. After their creation, the smart contracts are written in the blockchain and are therefore final and unchangeable. Hence, the corresponding part of the program has to be checked carefully in advance regarding the sequence of events. To use the contained source code, when using a smart contract, a request (transaction) is sent by a network participant with appropriate parameters to the address of the smart contract. From this point on, the automation is the responsibility of the program code, which is processed accordingly.

Closely related to the topic of smart contracts is the use of oracles. Execution in an appropriate runtime environment with appropriate APIs (e.g. REST, “Representational State Transfer”) on the blockchain ensures the execution security of smart contracts. This runtime environment ensures that the program code does not gain unauthorized access to local resources, binds it to the blockchain for reading and writing, and guarantees the deterministic execution of smart contracts. Furthermore, the results (transactions) of a smart contract are

stored decentrally in the blockchain and are therefore tamper-proof and irreversible. For the interaction of a smart contract with the outside world, i.e. with systems that are not included in the blockchain, so-called oracles are used. Depending on the application, these can be software, hardware, inbound, outbound, or consensus-based oracles [20].

3. SYSTEM ARCHITECTURE

The starting point for the conceptual design are production systems, i.e. machines in the field that generate process data, as shown in Fig. 1. A 3D printer is used for this work in particular. Here there are the data sources "machine control" which provides position, speed, and temperature data as well as machine status information via OPC UA, and "sensor nodes" whereby the latter was added to the existing systems through retrofit actions [21]. This consists mainly of sensor nodes based on the ESP32 microcontrollers as well as temperature and MEMS acceleration sensors.

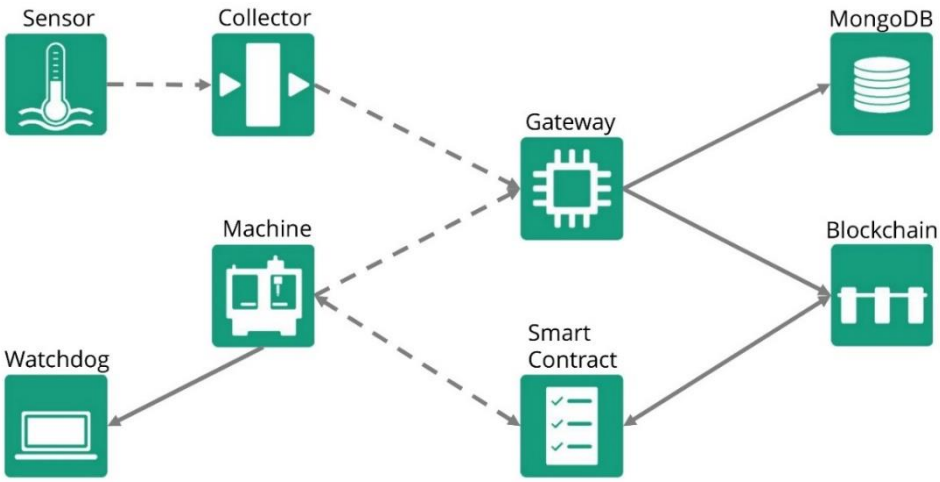

Fig. 1. Architecture for the implemented system

Since the sensor nodes are not directly integrated into the OPC UA address space, they first send the data stream via MQTT to a broker assigned to them: the Collector. Collectors provide this data to further services via an own OPC UA server. A gateway orchestrates the various data streams from the collector and the OPC UA server of the machine control system. Due to the high frequency of the data, which varies depending on the source, especially with regard to transaction data, it is necessary to use a data buffer. The Kafka framework is used for this purpose. In addition, the Gateway´s main function is to ensure the connection between different protocols. The gateway holds a corresponding client of a database system as well as a client and an oracle for the distributed ledger services. The latter provides the smart contract program code with the relevant data – for demonstration purposes, the continuously recorded temperature value on the machine. The smart contract compares this with a specified limit

value and returns a “START” or “STOP” signal depending on the content, which is returned to the machine and triggers the corresponding action.

To obtain an instance that tracks the status of the machine independently of other hardware and software, a so-called watchdog is implemented. In this way, a change in the status of the machine can be verified and communicated to subsequent services.

4. IMPLEMENTATION

The edge device consists of a single-board computer with Intel Pentium Quad-Core CPU, 4GB RAM, 64GB flash storage, and two Ethernet as well as a 2.4GHz Wifi interface. Ubuntu 20.04 LTS is used as an operating system. For the upcoming test purposes, a 3D-printer with secondary drives and additional components was simulated. This has the advantage that different configurations can be tested without danger to man and machine. Furthermore, the Kafka framework for data processing, the MongoDB for data storage, a MQTT Broker and a Geth(Go Ethereum)-based Blockchain were installed on each of the devices.

The sensor nodes based on the ESP32 microcontroller deliver a data stream via Wi-Fi through the MQTT protocol to the next instance and provide context information such as temperature values and acceleration data. The latter in particular are generated at an extremely high density. For this reason, the data is then pre-processed, in the sense that the data is combined without losing the basic information. Especially for oscillations, a Fast Fourier Transformation (FFT) is useful to extract valuable knowledge and reduce the amount of data at the same time. Since an FFT must be performed over a discrete amount of data, a ring buffer is programmed in which the acceleration values are buffered.

By modelling the machine controller as an OPC UA object, all variables can be directly read out there, so that the context data must now be added. By combining the different data streams, a data fingerprint of a cyber-physical production unit is created. This is achieved by the Collector creating another OPC UA object containing all variables from the controller and sensor network. From this point on, the data is buffered using the Kafka stream API to divide it into packets of equal size. For structured storage, JSON files are created according to the Production Performance Management Protocol Specification [22].

These are named after the syntax “MachineID_timestamp” and persisted in a MongoDB replica set via an appropriate interface. Every collector in the peer-to-peer network participates in this set. This contains all relevant information: A unique identification of the source “device” is ensured via “deviceID” = name of the collector, “source” = name of the source if it is data from the controls or a sensor node (“sensorNode”), and via “machineID” = name of the machine. The next element “measurements” contains the actual data. Besides the mandatory “ts” (timestamp), only “temperature” and “velocity” are provided as examples. For the software implementation, a SHA2-256 algorithm standardized by NIST 2015 [23] is used in the first programming step. An implementation with a SHA3-512 is currently being discussed and tested for feasibility. Now the hash values are distributed to all nodes in the private blockchain. This kind of data storage makes it possible to detect

manipulations of the recorded data in the MES, as well as data reconciliation. The hash value serves as the primary key in the MES, which is unique due to its collision resistance.

For the software implementation, a SHA2-256 algorithm standardized by NIST 2015 [23] is used in the first programming step. An implementation with a SHA3-512 is currently being discussed and tested for feasibility. Now the hash values are distributed to all nodes in the private blockchain. This kind of data storage makes it possible to detect manipulations of the recorded data in the MES, as well as data reconciliation. The hash value serves as the primary key in the MES, which is unique due to its collision resistance.

Hash functions are one-way functions, which map a text of arbitrary length to a text of fixed length and are collision resistant if the result space is large enough:

$$h = H(m)$$

where: h – value of a hash function, H – hash function (e.g. SHA, MD5) m – message or data

$$m_1 \neq m_2 \wedge H(m_1) = H(m_2)$$

This means that collisions where two different texts map to the same hash value (see equation) are almost impossible. This mapping of a large amount of data to a string of strict length allows comparisons to be made quickly and easily and manipulations to be detected. At the same time, it is impossible to draw conclusions from the hash value to the actual data, thus creating confidentiality. In contrast, it is easy to use the data to determine the hash value to confirm data integrity.

For the automation of the 3D printer by smart contracts, the communication with the blockchain is done via an oracle written in Python3 based on the library “web3.py”. After the smart contract has been deployed, its address is known. This allows it to be called with different input values and the return to affect the corresponding controlling variable of the 3D printer.

As output, the respective start and stop time and the corresponding temperature are given. The threshold for the stop trigger was arbitrarily set to 350 K and by a provided hysteresis the start trigger is at a value of 340 K. Referring to the example illustrated in Section 1, this means that above the set temperature limit, an unwelcome influence on the manufacturing quality occurs, which must be transparently protocolled.

By using the Smart Contract, the entire printing process can thus be logged, with parameter deviations leading to an automatic machine stop.

5. SUMMARY AND FURTHER STEPS

For the automation of data-controlled services on machine, machine safety must be guaranteed at all times. From the point of view of machine safety, there is currently no way of documenting the correctness of the work in an automated and tamper-proof manner. The scenario of service automation outlined here makes it possible to realize aspects of service fulfilment through the use of distributed ledger technology despite increased safety

requirements in mechanical and plant engineering. In this context, possibilities are outlined how the connectivity of industrial cyber-physical production systems and distributed ledger technologies can be used to realize secure data exchange and service automation via smart contracts, thus enabling a secure interaction of production plants with their environment. Necessary and established security measures can be taken into account by appropriately formulated smart contracts and the service used by the machine tool can be initiated.

In addition to the goal of traceability of process data from a manufacturing process, the integration of continuous system monitoring (e.g., predictive maintenance) into the architecture is also possible. This connection can increase the degree of autonomy of production systems.

One advantage of this implementation is its high adaptability. By using the most extensive and universal interfaces possible, such as the OPC UA protocol and the creation of an interface to the MQTT standard, the components behind it can be easily exchanged. This means that any sensor unit as well as machine with variables controllable via OPC UA can be integrated. Thus, it follows that the implemented case of temperature-dependent control of the digital control switch of a machine demonstrator does not limit the use of the control system. However, from the lessons learned from the implementation, it can be stated that universality is countered by high latency for control instructions through smart contracts compared to simple control software. Thus, the use of this trustworthy system in critical use cases is questionable and should rather be sought in the area of time- and safety-uncritical applications, as in the implemented case. Rather, the advantage created by the use of DLT lies in the increase in transparency and control of the control instructions carried out and its use as a secure interface to applications outside the production system. Users of a system implemented in this way have the option, for example with regard to quality management, of accessing information that is certainly unchanged.

The next step is to realize the automation of all relevant system variables of a CPPS using smart contracts in connection with an online service platform, for example, based on the model of a vending machine. Furthermore, such a scaling hybrid blockchain network must be implemented in a real production environment and its stability tested.

ACKNOWLEDGEMENTS

This Paper has been financed by BMWi Program "Smarte Datenwirtschaft" – Project "AUDlo-Auditlösung für ML-basierte, datengetriebene Dienstleistungen" Reg.-Nr.: 01MD19005, Germany. It was supported by the Federal Ministry for Economic Affairs and Energy (BMWi) based on a decision taken by the German Bundestag. This work was also supported by the Fraunhofer Internal Programmes under Grant No. MEF 836273.

REFERENCES

[1] BÖHLER T., 2020, *So Schafft Einzelteilrückverfolgung Mehr Transparenz*, https://www.produktion.de/technik/so-schafft-einzelteilrueckverfolgung-mehr-transparenz-262.html, (14.05. 2020).

[2] FEDKENHAUER T., *Datenaustausch Als Wesentlicher Bestandteil der Digitalisierung*, https://www.pwc.de/de/digitale -transformation/studie-datenaustausch-digitalisierung.pdf, (14.05.2020).

[3] BOTTHOF A., HARTMANN E.A., 2015, *Zukunft der Arbeit in Industrie 4.0*. Springer Berlin Heidelberg, 99.

[4] GSMA HEAD OFFICE, 2020, *Opportunities and Use Cases for Distributed Ledger Technologies in IoT*, https://www.gsma.com/iot/wp-content/uploads/2018/09/Opportunities-and-Use-Cases-for-Distributed-Ledgers-in-IoT-f.pdf, 14.05.2020.
[5] NEUGEBAUER R., 2018, *Digitalisierung: Schlüsseltechnologien für Wirtschaft und Gesellschaft*, 154f, Springer.
[6] WITT E., ANTON C., 2020, *Additive Fertigung. Entwicklungen, Möglichkeiten und Herausforderungen Stellungnahme*, Deutsche Akademie Der Naturforscher Leopoldina e.V., Halle (Saale).
[7] KLAHN C., MEBOLDT M., HÖFFKEN S., 2020, *So Funktioniert Qualitätssicherung in der Additiven Fertigung*, https://www.mission-additive.de/so-funktioniertqualitaetssicherung-in-der-additiven-fertigung-a-887397/, (01.11. 2020).
[8] NEUGEBAUER R., 2018, *Digitalisierung: Schlüsseltechnologien für Wirtschaft und Gesellschaft*, 197. Springer.
[9] GEBHARDT M., WEGENER K., 2013. *Temperatureinfluss auf Werkzeugmaschinen*, Machine Tools and Machinery (Manufacturing Technology), ETH Zürich, 59–63.
[10] KAFKA A., 2020, *Kafka 2.4 Documentation*, https://kafka.apache.org/documentation/#introduction, (28. 3 2020).
[11] REINERO B., 2017, *Transitioning from Relational Databases to MongoDB – Data Models*, https://www.mongodb.com/blog/post/transitioning-from-relational-databases-to-mongodb, 27/4.
[12] GÖTZ C., 2020, *MQTT: Protokoll für das Internet der Dinge*, https://www.heise.de/developer/artikel/MQTT-Protokoll-fuer-das-Internet-der-Dinge-2168152.html?seite=all, 08.05.2020.
[13] INRAY INDUSTRIESOFTWARE GMBH, 2020, *Was ist OPC UA?*, https://www.opc-router.de/was-ist-opc-ua/, 28/3.
[14] VDMA, FRAUNHOFER IOSB-INA, 2017, *Industrie 4.0 Kommunikation mit OPC UA*.
[15] MANDL P., 2018, *TCP und UDP Internals*, Springer Vieweg.
[16] RED PULSE, *All Blockchain are DLTs but not all DLTs are Blockchain*, https://www.redpulse.com/insight/20190601/all-blockchain-are-dlts-but-not-all-dlts-are-blockchain--e14df7e07f, (09.05.2020).
[17] ZHENG Z., XIE S., DAI H., CHEN X., WANG H., 2017, *An Overview of Blockchain Technology: Architecture, Consensus and Future Trends*, IEEE 6th International Congress on Big Data.
[18] LEMME G., LEMME, D., NÖLSCHER K.A., IHLENFELDT S., 2020, *Towards Safe Service Ecosystems for Production for Value Networks and Manufacturing Monitoring*, Journal of Machine Engineering, 20/1, 117–126.
[19] SZABO N., 1997, *The Idea of Smart Contracts*, Satoshi Nakamoto Institute, https://nakamotoinstitute.org/the-idea-of-smart-contracts/.
[20] BOGENSPERGER A., ZEISELMAIR A., HINTERSTOCKER M., 2018, *The Blockchain Technology – a Chance to Transform the Energy Supply?*, Report section technology description, 46f.
[21] LEMME G., NÖLSCHER K.A., 2019, *Wireless Sensor Network for Retrofitting Production Systems*, 18 Fachgespräch Sensornetze der GI/ITG Fachgruppe, Kommunikation und Verteilte Systeme, Magdeburg.
[22] ECLIPSE, 2020, *Production Performance Management Protocol Specification*, https://www.eclipse.org/unide/specification/v3/machine-message#messageDetail, (14.05.2020).
[23] PRITZKER P., GALLAGHER P.D., 2015, *SHA-3 Standard: Permutation-Based Hash and Extendable-Output Functions*, Laboratory National Institute of Standards and Technology.