LEDGER
ledgerjournal.org

ISSN 2379-5980 (online)
DOI 10.5195/LEDGER.2019.175

PROCEEDINGS ARTICLE

Robotchain: Using Tezos Technology for Robot Event Management

Miguel Fernandes*† Luís A. Alexandre‡

Abstract. Robots are important equipment in the modern day factory environment. To maintain and improve factory productivity, ledgers containing robotic actions may be used to identify possible bottleneck points in an assembly line or to serve as a record of in unintentional behaviours, be it of a malicious nature or not. In this paper we present Robotchain, a possible solution using blockchain technology, that prevents unwanted changes in a robotic action ledger and provides a way to use the said ledger in order to aid in production efficiency or other management requirements. This paper also presents an initial experimental study of the Tezos blockchain in order to understand the challenges related to using its advanced blockchain technology for the Robotchain implementation.

1. Introduction

This paper presents Robotchain, a private blockchain aimed for robotic enabled factories as a method for keeping a ledger of each and every action performed. This ledger is useful in order to understand production line performance, where to find possible points of improvement, and where to find possible faulty or under-performing robots. Said under-performing or faulty robots are a hindrance for the processing line, since they are high costly machines that have to pay back their investment in a short amount of time. With a cryptographically-secured ledger (*i.e.*, a blockchain), posterior changes to the ledger in order to hide under-performing robots are hard to perform, requiring vast amounts of resources. Robotchain can also be of use when there are accidents at the factory and there is the need to understand what went wrong, or who is at fault.

Systems for production monitoring, such as presented in Qu *et. al.*, "Online Monitoring of Manufacturing Process Based on autoCEP" (2017),[1] store information to use data mining methods, or the system presented in Snatkin *et. al.*, "Real Time Production Monitoring System in SME" (2012),[2] where the information is stored in a database, cannot store information in a way that cannot be tampered with, as opposed to our proposal. Blockchain technology is desired in order to prevent malicious entities from modifying the ledger, preventing manipulation of information coming from the robots.

A series of guidelines developed by the US Department of Homeland Security Science and Technology Directorate and provided by NIST are proposed to assess whether a given application

* bc1qn83aey30m6k2f0sz9nt4p3tvvw79uz5r2v2s5p

† M. Fernandes (ivo.fernandes@ubi.pt) is a researcher in the Departamento de Informática at the Universidade da Beira Interior and Instituto de Telecomunicações in Covilhã, Portugal.

‡ L. A. Alexandre (luis.alexandre@ubi.pt) is a full professor in the Departamento de Informática at the Universidade da Beira Interior and Instituto de Telecomunicações in Covilhã, Portugal.

should use a blockchain or not.[3] The first point to consider is if the application needs a consistent and shared data storage. In our case the answer is yes since the robotic events to be registered can not be altered and that information is to be shared among the involved stakeholders: the robot manufacturers that supplied robots to the factory and the factory administration. The robot manufacturers only access the information inside the factory, and only when there is the need to clarify a robot malfunction, but nonetheless, there is the need to access the data in a way that is not "filtered" by the factory employees. It is clear that there are several entities involved, and this answers positively the second point to consider, which is whether more than one entity needs the ability to contribute data. The third important aspect is if the records once written are to be updated or deleted. In our case the answer is no: the information must not be altered in any way. Regarding the fourth point, the issue of storing sensitive information, again, our use case indicates no need to store such data, as only robot IDs and events are stored in the blockchain. The fifth point, the issue of who should control the data storage, is central to this application. The factory would want to be in charge of the data, since it is produced on its manufacturing process, but at the same time, the robot manufacturers want to be able to prove their equipment's activities without the possibility of tampering by the factory (or other manufacturers). In regards to this, the use of a blockchain is again the solution. The final question is if we need a tamper-proof log of all recorded events, to which the answer is obviously yes, since that is the whole idea. Given the preceding discussion, we can conclude that our application scenario fits the type of situation where a blockchain-based solution is appropriate.

This project will use Tezos technology in order to develop the solution.[4] Since it will work in a contained and limited environment, such as a factory, this blockchain is deployed over a private LAN (Local Area Network), and since robots can produce a large number of actions in a small amount of time, performance for the blockchain is a priority. In this paper we present preliminary tests on a sandbox Tezos blockchain in order to understand possible limitations and positive points of using Tezos technology for this use case.

2. Related Work

There are already some proposals to integrate blockchain technology with robots, but none that cover the use cases that Robotchain contemplates. We detail these in the present section.

Castelló Ferrer, "The Blockchain: A New Framework for Robotic Swarm Systems" (2016), presents how blockchains can improve robotic swarm systems by solving some existing issues.[5] Those issues are data confidentiality, distributed decision making, the ability to work in different and dynamic environments without changes to the control program, and a way to ensure safety and legal responsibility for the robotic nodes in the swarm in order to be 'integrated' with human society.

Strobel *et. al.*, "Managing Byzantine Robots via Blockchain Technology in a Swarm Robotics Collective Decision Making Scenario" (2017),[6] presents a proof-of-concept method that uses blockchain-based smart contract technology in order to improve the security of the robotic swarm to improve the stability of the swarm coordination mechanisms and expel Byzantine members from the swarm. This concept is also studied for its performance in decision making regarding the presence or absence of Byzantine robots. Byzantine Fault tolerance is the concern for fault tolerance in distributed computer systems where components may fail or be unreliable.

ISSN 2379-5980 (online)
DOI 10.5195/LEDGER.2019.175

Danilov *et. al.*, "Towards Blockchain-Based Robonomics: Autonomous Agents Behavior Validation" (2018),[7] presents a model for a kind of trading marked named *robonomics*. It is focused on agent-based systems, where the behaviour is described as nondeterministic finite state automata, presenting a Model Checking verification technique in order to detect and filter malfunctioning agents. The validation technique can be implemented on a consensus protocol or as part of a blockchain decentralised application. As a real live test, a prototype implementation of Duckietown with moving robots is provided, following a set of instructions related to movement.

In Castelló Ferrer *et. al.*, "RoboChain: A Secure Data-Sharing Framework for Human-Robot Interaction" (2018),[8] a learning framework, called *RoboChain*, is presented. It attempts to solve privacy issues related to using personal information with blockchain technology, sharing data and machine learning models, allowing multiple robotic units to work at different places, sharing their data and their knowledge. It uses the latest technologies related to blockchains and machine learning in embedded devices such as low-cost robotic units. Since this approach assumes a situation where the participants are private entities (there is no public access), a level of trust between said parties is assumed and as such, the presence of a 'malicious entity' is not considered, but it still provides a way to verify the integrity of the interactions and learning in the blockchain.

Hasan *et. al.*, "Chained of Things: A Secure and Dependable Design of Autonomous Vehicle Services" (2018),[9] presents a blockchain framework for a secure ride-sharing service between autonomous vehicles and passengers, using a blockchain as a communication mechanism that is dependable and trustworthy.

Kambria is a project by Kambria International that will attempt to create an innovation and collaboration platform where the objective is improve development speed and adoption for robotic technologies.[10] It does this by preventing the "reinvention of the wheel" and allowing users to share their knowledge, be it code or schematics, and also allowing companies to tap into the collective developer knowledge. The developers are given an incentive with the integration of blockchain and crypto-economics, since it provides economic incentives to contribute to the platform. There is also the intent to "punish" developers that "defect," by using a game theory technique named "Grim Trigger."

Considering the above, our proposal is unique due to the fact that it is a robotic event ledger with smart contracts, having the capability of robot monitoring and fine tuning, as explained in the next section in more detail.

3. Our Proposal: The Robotchain

3.1. The Goals—Robots are becoming ever more important in modern factories. Currently it is a difficult task to keep track of every single action performed by every robot, in order to understand where possible bottlenecks are present or which robots need tuning, maintenance, or even replacement.

Our proposal contemplates the use of blockchain technology in order to solve the problem of keeping accurate immutable records of robotic actions in a factory environment. A public-access blockchain is not desired, since the factory environment is a private environment and, as such, management does not allow outside access to its internal manufacturing information.

In the case that concerns us in this paper, we are dealing with a set of robots from multiple manufacturers that are working at the same factory. We need for all the robot manufacturers and

ISSN 2379-5980 (online)
DOI 10.5195/LEDGER.2019.175

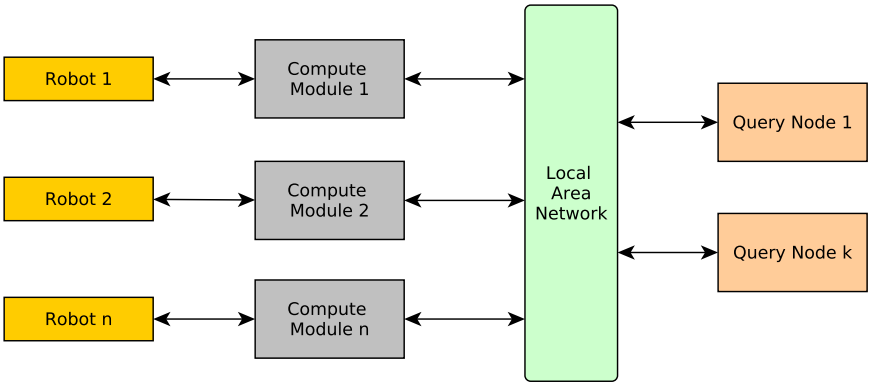

Fig. 1. A schematic view of Robotchain.

also the factory management to trust the event records, in the case there is an accident and fault has to be determined. The event records, which will be stored in a blockchain, can be used for further goals such as understanding and improving manufacturing productivity.

Currently, there is no standard for keeping this type of ledger since robots from different manufacturers may keep records using different formats, if they are kept at all. As such, this proposal fills a void in robotic event registration and tracking. With the addition of smart contracts, there is also the possibility of using artificial intelligence on the blockchain to optimize the control of robots or detect certain abnormal situations (sub-performance or pre-failures).

Figure 1 presents our proposal in a schematic way. Each robot is connected to a computation module, and this connection is bidirectional in order to receive information from the robot to feed the blockchain and allow the blockchain, via smart contracts, to change the robot's behaviour. In a first version of this concept, the smart contracts will not be implemented, and as such, the connection between the robot and the computation module will initially be unidirectional.

The use of computational modules is to ensure a uniform input into the blockchain, as different robots may need different connection interfaces. It also ensures that the robots are not negatively affected with additional software running, that may cause degraded performance or other unforeseen consequences.

In addition, there can be query nodes connected to the blockchain network in order to query it for information. These are important for understanding possible production line bottlenecks, or improving management understanding of the factory without directly interfacing with the robotic units. The main concern is the high transaction volume that the various networked robots will produce. Also important is the fact that Robotchain may not impact in any form the performance of the robots.

3.2. Using Tezos—Goodman, "Tezos—A Self-Amending Crypto-Ledger" (2014),[4] presents a self-amending crypto-ledger implemented in *OCaml* called Tezos. Instead of using a genesis block or hash, it starts with a seed protocol, where this protocol can be amended in order to replicate other blockchains.

The main feature of this blockchain is the fact that it implements a protocol that can adapt itself by transforming a Context Object. These amendments work over cycles, which take about three months and are suggested by a submission to the chain. Stakeholders may vote for these

ISSN 2379-5980 (online)
DOI 10.5195/LEDGER.2019.175

amendments, and, if they are accepted, they are first inserted into a *testnet*. After that, a second confirming vote is made. If the second vote is successful, the amendments are integrated into the the main protocol.

These amendments are considered a positive point due to the fact that this allows the community to enact changes in the blockchain, in order to improve it, similar to a political system, preventing blockchain hard forks, which are a radical change that results in divergence from the already-created chain.

In addition to this, the fact that Tezos is open source allows us to adjust it to our purposes, mainly by trying to improve transaction speed, and possibly reduce complex steps in order to make it work on cheap computational units. Another positive points for using Tezos blockchain technology is an increase of security with respect to the manipulation of the ledger. Also, smart contracts will be proven correct, giving an additional layer of trust in the way the system is implemented. In comparison to other mainstream protocols, the self-amending feature present in the Tezos network and the verifiable security using the OCaml language are the distinguishing points provided by this network. The self-amending feature provides a way to upgrade the blockchain network, such as fine-tuning the consensus algorithm, or other algorithms present without creating a hard fork on the already existent network, and the fact that the code is proved correct is of extreme importance when dealing with high cost equipment that can condition the operation of a factory.

4. Experiments

4.1. Experimental Setup—For usage of the Tezos blockchain, experiments were made in order to understand the possible changes that are needed in order to use this system in the desired context.

Since Robotchain uses a private blockchain, contained inside a factory, unable to be accessed by outside means, the preliminary experiments were made using a Virtual Machine, using the provided sandbox configurations.

This virtual machine consists of 4 threads of an i7 CPU 960 @ 3.20GHz, leaving the other 4 threads for the host system, 8 GB of RAM, 20 GB of hard drive. It is virtualized a with KVM Hypervisor.

For performance measurement, several metrics were chosen, such as: time per transaction, number of crashed nodes, and blocked transactions. These metrics were chosen due to the importance of transaction speed, since the blockchain needs to handle the throughput of the robots, either the time it takes for each transaction and if the transaction actually occurs. Stability of the network is also considered important.

4.2. Trial Description—We conducted experiments with different "blocks per cycle" parameters, where the default value is 8, and the alternative tested value was 4, with a different number of nodes, (5, 10, 50 and 100) and each experiment consisted of five trials. For the 50 and 100 nodes, the experiments are considered stress tests to the network rather than stability or functionality tests, due to the fact that we are considering 50 or 100 programs running at the same time, under only 4 CPU threads, a setting which is not meant to be used in a real world scenario.

Each trial was run in the following manner.

N node instances are initialised, with connections from each node to all the other nodes.

ISSN 2379-5980 (online)
DOI 10.5195/LEDGER.2019.175

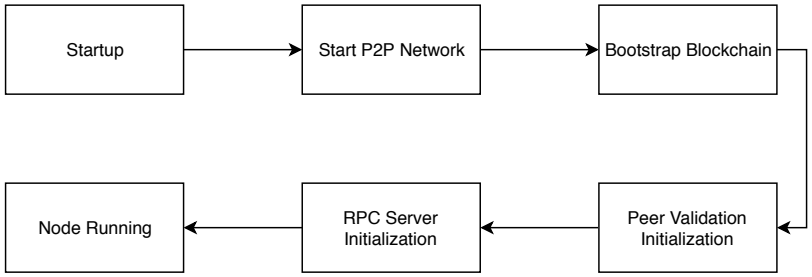

Fig. 2. Steps that occur during node initialisation.

These instances are created with the provided sandbox scripts, with the modification of the original 9 node limit to a maximum of 100 nodes.

The node initialisation process is described in Figure 2, where the bootstrap is the act of loading the information present locally and synchronizing it with the other nodes. The RPC Server present in the node is the interface used by the client processes to act in the blockchain.

Following the node initialisation, the sandbox-provided protocol parameters are used on the network, five bootstrap accounts are loaded, each with credit of 4000000 *tez*, where bootstraps one and two are used for transfers, and bootstraps three, four, and five are used as bakers. Bootstrap five is also nominated as a Delegate.

Rather than using the provided sandbox scripts for the baker and endorser nodes, two different choices for running the nodes were tested. They were either as standalone clients, where they bake or endorse *ad eternum*, or a request is made, via the regular client, to bake or endorse a block. The latter approach was selected in order to have a finer control of the resources used by the virtual machine.

As such, a random node is selected and accounts three to five, in parallel, bake and endorse a block. Also, in parallel to this, transfer requests are made, on random nodes and of random amounts of *tez*, from account one to account two and account two to account one. The time that each transaction takes is saved for statistical purposes. A timeout of 15 seconds for each transaction was defined. If the 15 seconds elapse without the transaction being successful, it is considered a failure, which is also accounted for in the statistics.

5. Results and Discussion

Tables 1 and 2 present the transactions statistics retrieved from examining the execution logs, and Figure 3 contains the average for transaction time as a function of the number of nodes, for both 4 blocks and 8 blocks per cycle.

The total transaction time is measured in seconds, without considering transactions that timed out, and the average transaction time is the average number of seconds that a successful transaction took.

The provided sandbox scripts only allow connections inside the same host. This setup has the advantage of not having a network delay.

For both 4 and 8 blocks per cycle, when using only 5 nodes, there are no timed-out transactions. These start to appear for experiments with 10 or more nodes. Note that the sandbox original setup parameters were limited to 9 nodes, and as such, we are running these experiments outside

ISSN 2379-5980 (online)
DOI 10.5195/LEDGER.2019.175

Table 1. Transaction statistics for the trials in the experiments with four blocks per cycle. Using a virtual machine with four threads allocated.

Number of nodes	Total transaction time	Avg. transaction time	Timed-out transactions
5	1003.69	1.00	0
5	985.77	0.99	0
5	988.20	0.99	0
5	986.77	0.99	0
5	987.81	0.99	0
10	934.68	1.14	179
10	1871.82	1.88	5
10	1653.63	1.66	1
10	1851.15	1.86	5
10	1635.62	1.67	22
50	2480.74	3.21	226
50	3408.44	3.90	127
50	3442.78	4.09	158
50	3027.17	3.52	139
50	3189.37	4.07	216
100	1715.61	4.93	652
100	1286.90	5.15	750
100	672.81	4.13	837
100	965.73	1.05	77
100	1083.11	4.25	745

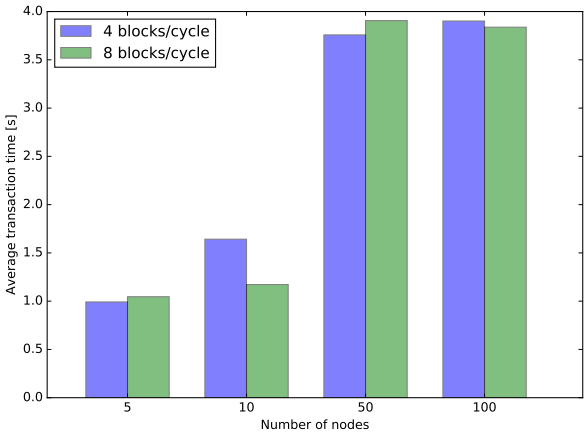

Fig. 3. Average transaction time as a function of the number of nodes.

ISSN 2379-5980 (online)
DOI 10.5195/LEDGER.2019.175

Table 2. Transaction statistics for the trials in the experiments with eight blocks per cycle (the default value), using a virtual machine with four threads allocated.

Number of Nodes	Total Transaction time	Avg. Transaction time	Timed-out transactions
5	994.3	0.99	0
5	990.5	0.99	0
5	1141.8	1.14	0
5	1120.8	1.12	0
5	989.6	0.99	0
10	995.0	1.01	20
10	1127.8	1.13	1
10	1357.1	1.37	9
10	1222.6	1.23	2
10	638.7	1.12	432
50	3322.6	3.92	152
50	3244.6	3.85	157
50	3203.0	3.90	179
50	2941.2	3.74	213
50	3306.1	4.12	198
100	1713.8	3.33	485
100	594.4	2.73	782
100	1666.6	3.89	572
100	3020.1	4.79	369
100	2230.0	4.45	499

ISSN 2379-5980 (online)
DOI 10.5195/LEDGER.2019.175

of the original specifications.

From the presented results we conclude that the change in the blocks per cycle from 8 to 4 did not produce any noticeable performance change. The justification for this is that the actions performed at the end of each cycle, namely the baker credit payout and backer staking, do not alter the overall statistics if done every 4 or 8 blocks.

As the number of nodes increase, we see a larger number of timed out transactions. As an example, running 100 nodes, 3 bakers and 3 endorsers, results in the total of 106 processes, and these processes are being handled by only 4 threads, meaning, each thread is processing 26 processes each, which may justify the observed decreased success rate. Similar analysis can be carried out for the other cases.

According to Figure 3, in the 5 node experiments we had a 1 second average transaction time which is still far from the desired transaction speed for our use case.

We did not observe any noticeable performance change with variation in the "block per cycle" parameter, due to the fact that this parameter does not produce significant work in the nodes.

6. Conclusion

In this paper we proposed a new blockchain tailored to registering robotic events in closed environments, such as factories, called Robotchain.

Our proposal is not just useful for guaranteeing an immutable event ledger that can be used for deciding which robot made particular actions in conflicting situations, but also allows for performance monitoring and even robot tuning by taking advantage of smart contracts, although these settings were not explored in the current paper.

The paper also contains exploratory experiments to gain insight on the critical factor of transaction time, using a sandbox implementation of a Tezos blockchain. The results point to the need for changes that enable at least two orders of magnitude speedup with regard to the current values.

As such, future work will be focused on improving transaction speed. Using Tezos blockchain technology is important due to the fact that it already has an implemented basis for verifiable security and smart contracts, allowing us to focus our efforts on improving the transaction throughput for our specific application scenario.

Acknowledgements

This work was partially supported by the Tezos foundation through a grant for project Robotchain.

Notes and References

[1] Qu, J., Li, S., Chen, J. "Online Monitoring of Manufacturing Process Based on autoCEP." *iJOE* **13.6** 22–34 (2017) http://www.online-journals.org/index.php/i-joe/article/view/6812.

[2] Snatkin, A., Karjust, K., Majak, J., Aruväli, T., Eiskop, T. "Real Time Production Monitoring System in SME." In *Proceedings of the 8th International Conference of DAAAM Baltic, INDUSTRIAL ENGINEERING, 19-21 April 2012, Tallinn, Estonia* (2012) http://innomet.ttu.ee/daaam_publications/2012/snatkin.pdf.

[3] Yaga, D., Mell, P., Roby, N., Scarfone, K. "Blockchain Technology Overview (NISTIR-8202)." *NIST: National Institute of Standards and Technology* (2018) (accessed 9 March 2019) https://csrc.nist.gov/publications/detail/nistir/8202/final.

ISSN 2379-5980 (online)
DOI 10.5195/LEDGER.2019.175

[4] Goodman, L. "Tezos - A Self-Amending Crypto-Ledger." (2008) Whitepaper (accessed 9 March 2019) https://tezos.com/static/papers/white_paper.pdf.

[5] Castelló Ferrer, E. "The Blockchain: A New Framework for Robotic Swarm Systems." *arXiv* (2016) (accessed 9 March 2019) http://arxiv.org/abs/1608.00695.

[6] Strobel, V., Castelló Ferrer, E., Dorigo, M. "Managing Byzantine Robots via Blockchain Technology in a Swarm Robotics Collective Decision Making Scenario." In *AAMAS '18 Proceedings of the 17th International Conference on Autonomous Agents and MultiAgent Systems* 541–549 (2018) https://dl.acm.org/citation.cfm?id=3237464.

[7] Danilov, K., Rezin, R., Kolotov, A., Afanasyev, I. "Towards Blockchain-Based Robonomics: Autonomous Agents Behavior Validation." *arXiv* (2018) (accessed 9 March 2019) http://arxiv.org/abs/1805.03241.

[8] Castelló Ferrer, E., Rudovic, O., Hardjono, T., Pentland, A. "RoboChain: A Secure Data-Sharing Framework for Human-Robot Interaction." *arXiv* (2018) (accessed 9 March 2019) http://arxiv.org/abs/1802.04480.

[9] Hasan, M. G. M. M., Datta, A., Rahman, M. A., Shahriar, H. "Chained of Things: A Secure and Dependable Design of Autonomous Vehicle Services." In *2018 IEEE 42nd Annual Computer Software and Applications Conference (COMPSAC)*. **02** 498–503 (2018) https://doi.org/10.1109/COMPSAC.2018.10283.

[10] Kambria: https://kambria.io.

SSN 2379-5980 (online)
DOI 10.5195/LEDGER.2019.175