Improving Railway Maintenance Actions with Big Data and Distributed Ledger Technologies

Roberto Spigolon[1], Luca Oneto[2], Dimitar Anastasovski[1], Nadia Fabrizio[1], Marie Swiatek[3], Renzo Canepa[4], Davide Anguita[2]

[1] Cefriel - Politecnico di Milano, Italy
{roberto.spigolon,dimitar.anastasovski,nadia.fabrizio}@cefriel.com
[2] DIBRIS - University of Genoa, Italy
{luca.oneto,davide.anguita}@unige.it
[3] Evolution Energie, France
marie.swiatek@evolutionenergie.com
[4] Rete Ferroviaria Italiana, Italy
r.canepa@rfi.it

Abstract. Big Data Technologies (BDTs) and Distributed Ledger Technologies (DLTs) can bring disruptive innovation in the way we handle, store, and process data to gain knowledge. In this paper, we describe the architecture of a system that leverages on both these technologies to better manage maintenance actions in the railways context. On one side we employ a permissioned DLT to ensure the complete transparency and auditability of the process, the integrity and availability of the inserted data and, most of all, the non-repudiation of the actions performed by each participant in the maintenance management process. On the other side, exploiting the availability of the data in a single repository (the ledger) and with a standardised format, thanks to the utilisation of a DLT, we adopt BDTs to leverage on the features of each maintenance job, together with external factors, to estimate the maintenance restoration time.

Keywords: Big Data Analytics, Distributed Ledger Technologies, Railway Maintenance Actions.

1 Introduction

Railway Infrastructure Managers (IMs) are responsible of operating the existing rail infrastructures. Maintenance is, without any doubt, one of the main task of this job [5, 3]: not properly maintained infrastructures are in fact more prone to failures, that in turn translate into disruptions of the normal execution of railway operations.

Maintenance operations are usually demanded to external contractors through framework contracts that guarantee the availability of specialized workers whenever there is a need, planned in advance or unexpected. Considering in particular the planned maintenance operations, the actual work is scheduled on specific time slots, where the train circulations can be modified or suspended without causing major disruptions. An empirical estimation of the time needed to perform the jobs is used to plan the scheduling of all the operations on the available time slots.

Moreover, to ensure that each maintenance job is performed in the correct way, that all safety measures are put in place, and all responsibilities are clearly identified, IMs employ standardised procedures to guarantee that each action is executed in a proper order by the responsible actor, leaving a trace of that execution. To fulfill such requirements, each step in these procedures must be performed leaving a legally valid record of which actions were performed, by whom, and with which authorisations; thus leading to a lot of signed papers sent between the various actors via registered letters, and to recorded phone calls. A process involving paper documents can be inefficient, leading to increased waiting times between each step in the workflow. Also having a direct access to maintenance data, to assess the current status of a specific job or to perform data analysis [9, 6], may not be straightforward.

In this context, BDTs and DLTs may bring a great benefit to the current management of maintenance jobs. From one side, the adoption of DLTs and smart contracts could enable the digitalisation of the process currently employed maintaining all the required features, like a tamper-proof record for the tracking of all decisions and executed actions [2], and potentially allowing the automated enforcement of contractual clauses. From the other side, the analysis of historical data about previous maintenance operations could enable the development of a prediction algorithm able to accurately estimate the restoration time for each maintenance job, thus leading to a better planning of the operations. Moreover, the DLT enacts the gathering of all the data on a single repository (the ledger) and with a standardised format, allowing the periodic retraining of the prediction engine: such operation could hardly be done with data stored in isolated silos with a noninteroperable format.

For this reason in this paper we propose an architecture able to merge DLTs, to automate the railways maintenance workflows, and BDTs for improving the decision of IMs in executing railway operations.

2 DLT-based Maintenance Management

The system described, currently under development, comprises two major components: a DLT peer-to-peer network and a prediction engine. The DLT network will be implemented using Hyperledger Fabric version 1.3[5]: a permissioned DLT, where only authorised peers may join the network [4]. The selection was conducted comparing the currently available solutions using a scoring model derived from the requirements of IMs; in particular we referred to the Italian Railway Network handled by Rete Ferroviaria Italiana (RFI) which defined a list of requirements.

Figure 1 shows the logical architecture of the network, based on Hyperledger Fabric components [1]. Each organization participating in the maintenance operations management scenario has its own peers and a Certification Authority (CA). Each CA acts as a Membership Service Provider (MSP) for its own organization, and provides digital certificates to its related peers. The network is globally administrated by the IM in its role as ecosystem leader. All the organizations instead share the membership service of the dedicated logical channel where all the peers are connected, allowing each of them to add their own peers to the channel. Both the ledger and the chaincodes (smart contracts in Hyperledger Fabric) are replicated on every peer connected to the channel, providing

[5] `www.hyperledger.org/projects/fabric`

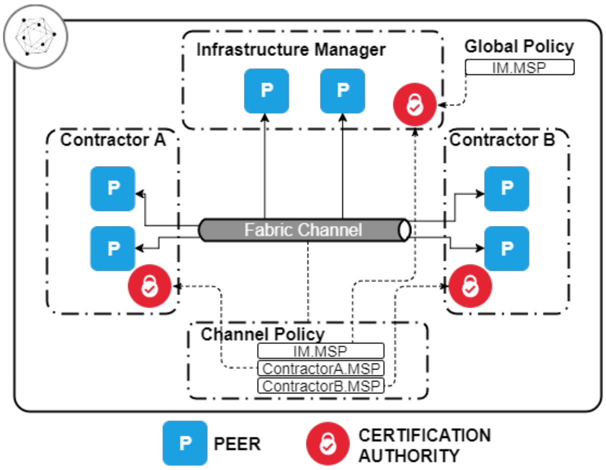

Fig. 1. A logical view of the Hyperledger Fabric network: each organization has its own peers authorized by the respective Certification Authority and connected to a single Fabric channel. Please note that the number of peers is not relevant in this view.

redundancy of the data. The ordering service will be provided by a single orderer on a first testing phase, to be later developed to a crash tolerant Kafka cluster [1] on a later stage.

In our scenario, confidentiality of the data is not a critical requirement as the IM has a strong interest that all the process is transparent within all the authorized network participant. Nevertheless, Hyperledger Fabric v.1.3 enables the definition of Private Data within the same channel, to ensure that confidential data between two parties are not shared with the other participants: this is done using private databases separated from the main ledger, only broadcasting hashes of the private transactions on the main ledger. In addition, the network could easily support other scenarios and applications via the creation of different channels between the organizations, ensuring the complete separation of the respective ledgers.

The prediction engine is built as a separate component deployed outside the DLT network as it is not possible to implement it as a chaincode inside Fabric: the prediction engine needs to be able to automatically retrain (and therefore modify) the prediction algorithm, but chaincodes can only be updated manually. The interconnection between the two components will be developed through two REST APIs, as depicted in Figure 2.

Once a maintenance operation is about to start, an operator has to officially state it committing a transaction to the DLT, where she inserts all the exogenous data, like the current weather condition, needed to the prediction engine. The chaincode therefore calls *Predict*($\underline{X}$), where $\underline{\text{X}}$ stands for the required data for the prediction, getting in return the estimated restoration time and the tree model used to estimate it, that will be recorded as well on the ledger. It is important to note that the prediction algorithm does not need to retrieve additional data through external sources, since it gets everything from the chaincode. This is required to avoid non-determinism. Indeed, considering that the chaincode is executed by all the endorsing peers independently at potentially different times,

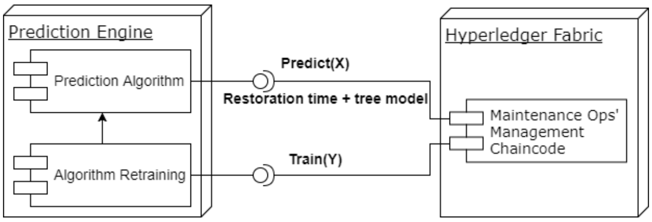

Fig. 2. Interconnection between the Hyperledger Fabric network and the Prediction Engine.

retrieving data from external sources could change the results of each execution, since there is no control on external data, preventing the consensus from being achieved. *Train(Y)* is responsible of retraining the prediction algorithm using all the data stored on the ledger so far. In this case, there is no risk of having non-determinism as that API does not return anything; of course, retraining actions should be performed only when there are no maintenance operation starting, to avoid changing the prediction algorithm when it is predicting the restoration time of a job, leading to the non-deterministic condition explained before.

3 The Prediction Engine

The prediction engine is in charge of estimating the time to restoration for different assets and different failures and malfunctions. The predictive model needs to take into account the knowledge enclosed into maintenance reports, exogenous information such as the weather conditions and the experience of the operators in order to predict the time needed to complete a maintenance action over an asset and to restore its functional status. Moreover, the model should be interpretable enough to give insights to the operators on which are the main factors influencing the restoration time, to better plan the maintenance activities. This information will help IMs to assess the availability of the network, by estimating the time at which a section block including a malfunctioning asset will become available again, and properly reschedule the train circulation.

For this purpose we have built a rule-based model which is able to exploit real maintenance historical data provided by RFI, the historical data about weather conditions and forecasts, which is publicly available from the Italian weather services, and the experience-based model currently exploited by the train operators for predicting the restoration time of planned maintenance. Then we implemented the Decision Tree [7] using the MLlib [8] in Spark, because of the huge amount of data available (approximately 1TB of data each year), and we deployed an infrastructure of four machines equipped with 32GB of RAM, 500GB of SSD disk space and 4 CPUs on Google Compute Engine[6]. We implemented the 10-fold cross validation for optimizing the number of points per leaf n_l by searching $n_l \in \{5, 10, 20, 50, 100, 200, 500\}$. All experiments have been performed 30 times to ensure the statistical robustness of the results. Table 1 reports:

Table 1. Quality of the models.

Int.	MAE	MAPE	PCOR
RFI			
Maint.	30.5	31.5	0.75
Our Proposal			
All	11.3±1.1	10.7±0.9	0.93±0.03
Maint.	8.1±1.0	7.8±0.7	0.97±0.03
Fail.	15.2±1.3	14.3±1.1	0.88±0.04

[6] https://cloud.google.com

- the error of the RFI model measured with the Mean Average Error (MAE), the Mean Average Percentage Error (MAPE), and the Pearson Correlation (PCOR) on the maintenance since no model for the failures is available to RFI;
- the error of the data-driven model measured with the MAE, MAPE, and PCOR on all the intervention, on the maintenance, and on the failures.

From Table 1 it is possible to note that the quality of our model is remarkably higher than the one of the RFI model.

4 Conclusions

The system described in this paper, built upon a permissioned DLT empowered with smart contracts and a prediction engine, permits the automated management of the highly regulated administrative workflow that each maintenance job has to deal with, while enriching it with the possibility to estimate the restoration time of each job, leading to a better planning of train disruptions. The main achievements of such system are twofold. The first one is to bring forward the digitalisation of the workflow currently employed ensuring integrity and non-repudiation of every action performed inside the workflow thanks to the native features of DLTs; permitting, as a consequence, the instant retrieval of the status of each maintenance job. The second one is to allow to better plan the maintenance operations thanks to the availability of an estimated restoration time for each job. Additionally, the system could be extended to enable the enforcement of contractual clauses (i.e. penalties for delays) via automatic execution of disputation procedures backed by evidence stored in the audit-proof ledger.

Acknowledgments This research has been supported by the European Union through the projects IN2DREAMS (European Union's Horizon 2020 research and innovation programme under grant agreement 777596).

References

1. Androulaki, E., Barger, A., Bortnikov, V., Cachin, C., Others: Hyperledger fabric: A distributed operating system for permissioned blockchains. In: EuroSys (2018)
2. Benbunan-Fich, R., Castellanos, A.: Digitalization of land records: from paper to blockchain. In: International Conference on Information Systems (2018)
3. Budai, G., Huisman, D., Dekker, R.: Scheduling preventive railway maintenance activities. Journal of the Operational Research Society 57(9), 1035–1044 (2006)
4. De Kruijff, J., Weigand, H.: Understanding the blokchain using enterprise ontology. In: International Conference on Advanced Information Systems Engineering (2017)
5. Farrington-Darby, T., Pickup, L., Wilson, J.R.: Safety culture in railway maintenance. Safety Science 43(1), 39–60 (2005)
6. Fumeo, E., Oneto, L., Anguita, D.: Condition based maintenance in railway transportation systems based on big data streaming analysis. In: INNS International Conference on Big Data (2015)
7. James, G., Witten, D., Hastie, T., Tibshirani, R.: An introduction to statistical learning. Springer (2013)
8. Meng, X., Bradley, J., Yavuz, B., Sparks, E., Venkataraman, S., Liu, D., Freeman, J., Tsai, D.B., Amde, M., Owen, S.: Mllib: Machine learning in apache spark. The Journal of Machine Learning Research 17(1), 1235–1241 (2016)
9. Thaduri, A., Galar, D., Kumar, U.: Railway assets: A potential domain for big data analytics. In: INNS International Conference on Big Data (2015)