

FUSION—Fog Computing and Blockchain for Trusted Industrial Internet of Things

Andrea Ceccarelli , Marcello Cinque , Christian Esposito , *Member, IEEE*,
Luca Foschini , *Senior Member, IEEE*, Carlo Giannelli , *Senior Member, IEEE*, and Paolo Lollini 

Abstract—The industrial Internet of Things (IIoT) is currently foreseen as a foundation to implement the Industry 4.0 vision. However, device heterogeneity and the need of integration and configuration exposes the industrial infrastructure to potential threats, such as black-hole, man-in-the-middle, and malicious configuration attacks. In this article, we investigate how to manage distributed trust information and to enable trusted configuration actions in the IIoT, by opportunistically intermingling blockchain with the software defined networking and container orchestration technologies. In particular, we focus on how the joint and coordinated adoption of such technologies can make technicians' interventions on industrial equipment both easier and more trusted. To this purpose, we present the design of a software architecture to simplify the management, configuration, and assessment of IIoT systems, and we discuss our experiences with the application of the proposed architecture in a railways use case.

Index Terms—Blockchain, fog/edge computing, industrial Internet of Things (IIoT).

I. INTRODUCTION

THE recent widespread use of sensor and actuator networks and the convergence of wireless/wired technologies into the so-called edge/fog computing are giving rise to the industrial Internet of Things (IIoT) [1], whose exemplification is in Fig. 1. IIoT refers to the application of the Internet of Things (IoT) paradigm to industrial environments, and it is nowadays considered the foundation to implement Industry 4.0 [2]. In a typical IIoT deployment, end devices, such as sensors and actuators installed on the field to control a given phenomenon or plant, are connected together. Moreover, often they interconnect through wireless channels, via a set of more powerful devices, generically indicated as edge/fog devices. Those edge devices host more advanced functionalities, and in turn can integrate

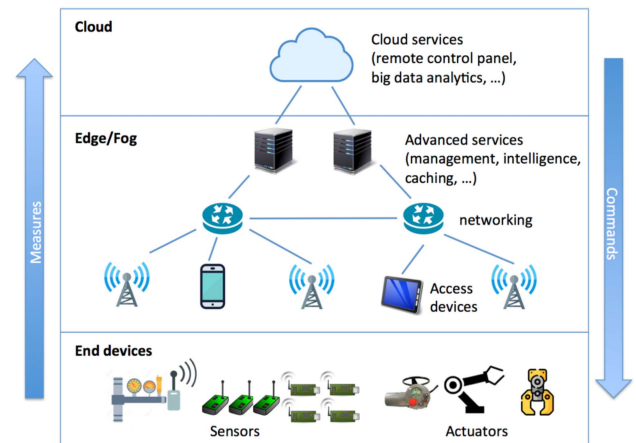


Fig. 1. High-level distributed architecture of IIoT.

advanced services from the cloud, similarly to the classical IoT cloud scenario. However, the IIoT scenario differs from the IoT cloud one due to its more stringent requirements in terms of heterogeneity, management, and trustworthiness.

Typical IIoT applications range from digital factories and product lifecycle management to the supervision and control of critical infrastructures, such as smart electric grids and smart transportation systems. In these applications, the need of trustworthiness and resiliency is urgent both in terms of safety and security, to prevent harms, life and money losses, and catastrophic failures.

The application of existing trust and security countermeasures to the IIoT is complicated by the large heterogeneity. Adopted devices range from resource constrained wireless microcontrollers to powerful supervision systems, robotic arms, and edge/fog devices, as shown in Fig. 1. Heterogeneity is also coupled with the large scale of IIoT devices and calls for innovative autonomous network management solutions to enable flexible and smart deployment and configuration of hardware/software/networking components based on application needs and constraints. Moreover, to further complicate this (realistic) scenario, several recent efforts are investigating how to effectively deploy distributed software defined networking (SDN) control planes atop heterogeneous edge/fog IIoT deployments [3].

The wide integration and reconfiguration of the hardware and software substrate, needed for the implementation and management of IIoT, increases the attack surface, exposing the

Manuscript received 26 February 2020; revised 4 July 2020 and 26 August 2020; accepted 27 August 2020. Date of publication 30 September 2020; date of current version 1 November 2022. Review of this manuscript was arranged by Department Editor K.-K. R. Choo. (*Corresponding author: Marcello Cinque.*)

Andrea Ceccarelli and Paolo Lollini are with the University of Florence, 50121 Florence, Italy (e-mail: andrea.ceccarelli@unifi.it; paolo.lollini@unifi.it).

Marcello Cinque is with the University of Naples Federico II, 80138 Napoli, Italy (e-mail: macinque@unina.it).

Christian Esposito is with the University of Salerno, 84084 Fisciano, Italy (e-mail: christian.esposito@unina.it).

Luca Foschini is with the University of Bologna, 40126 Bologna, Italy (e-mail: luca.foschini@unibo.it).

Carlo Giannelli is with the University of Ferrara, 44121 Ferrara, Italy (e-mail: gnnrcl@unife.it).

Color versions of one or more figures in this article are available at <https://doi.org/10.1109/TEM.2020.3024105>.

Digital Object Identifier 10.1109/TEM.2020.3024105

IIoT to advanced security threats. Examples are the addition of new sensor nodes or fog devices to perform black-hole and man-in-the-middle attacks, or the change of configuration of a device. For instance, the change of measurement rate could cause physical damage on the controlled cyber physical system (CPS) as in the famous case of Stuxnet [4]. While a plethora of solutions have emerged for IoT security, only a few have looked into malicious configuration issues. However, they often mandate the use of centralized authorities that do not scale with the size of the system [2]. In addition, even when those solutions are available, their offline assessment and prediction, in terms of trust properties, is often insufficient in highly complex dynamic systems as the IIoT [5], [6].

The adoption of IoT in the industrial environment and application is strongly increasing due to its growing economic importance in strengthening competitiveness and efficiency of the players in the industry sector. A study from the IoT platform particle [52] shows that the IoT industry may eventually have an economic impact of more than \$11 trillion by 2025, and IoT devices are mostly used for remote monitoring (78%), preventative maintenance (55%), and asset tracking (33%). However, the study also points out that IoT professionals struggle to manage IoT devices in the most effective and efficient way. For example, development teams have troubles to fix malfunctioning devices, install software updates, and perform other management operations in a secure and scalable way. This article proposes a solution tailored on the needs of IIoT professionals, able to cope with the security and scalability issues in IoT device management [53], so as to fully unleash the potential of IoT within industrial contexts. Moreover, the proposed solution opens up novel business opportunities by facilitating the outsourcing of maintenance activities to a contractor rather than hiring an in-house maintenance team, or supporting the predictive maintenance within the context of Industry 4.0. Such a so-called Maintenance 4.0 is extremely important for the industry sector, as it is expected to grow to \$6.3 billion by 2022 [54], and is calling for a new scalable and secure technological substrate, which can consist in the *fog and chains for trUSTed Industrial Internet Of ThiNGs* (FUSION) platform proposed in this article.

This article presents FUSION, a software platform, which has been designed to face the abovementioned challenges. FUSION promotes the convergence of *advanced network management* (based on SDN) and *configuration solutions* (based on cloud orchestration and container technologies), and it exploits *blockchain* as enabling technology for *trust and configuration management*. In particular, this article focuses on the specific goal of making both easier and more trusted the intervention of technicians on industrial equipment, by properly exploiting the abovementioned technologies in a joint and coordinated manner. To this purpose, we make the assumption that industrial equipment is connected to edge devices, the latter characterized by relevant computational and memory capabilities if compared with the former. Moreover, we limit the focus on the management of industrial equipment with the assumption that different technicians can operate on the same equipment at different times, and these technicians do not trust one another.

We observe that blockchain is currently foreseen as a disruptive technology by the IoT industry, and it is expected to

play a major role in managing, controlling, and securing IoT devices [7]. At the same time, despite the hype of the recent years, to the best of authors' knowledge, there are still no works addressing the specific challenges and threats of the IIoT scenario highlighted earlier. To overcome such limitations, FUSION originally proposes to integrate and intermingle blockchain with SDN and cloud container orchestration. The main idea is to treat network admin operations as transactions, traced, and validated through the blockchain, as currently done worldwide for crypto currencies. In particular, the FUSION platform pursues the following main elements of originality.

- 1) It investigates the use of blockchain technologies in the IIoT to perform decentralized trust management, to assure integrity, and to perform secure and resilient management and configuration operations.
- 2) It introduces the notion of FUSION edge smart nodes (FESNs) as computing nodes that can be flexibly reconfigured to act as end devices, access nodes, SDN nodes, fog nodes, etc., depending on application needs and exploring the adoption of the SDN approach in the IIoT to simplify the management of network resources.
- 3) It defines solutions for the continuous assessment of dependability and security properties in the IIoT, exploring tradeoffs to reduce the effort to specify complex models of the system, through model-driven engineering MDE, which are fed, at runtime, with data collected from monitors deployed on FESNs, to update the models and set/fine-tune specific model parameters resembling the current status of the system;
- 4) It describes our experience at applying FUSION to a realistic railway business use case. The idea is to show how FUSION can support physical infrastructures to enable innovative solutions to track and certify maintenance interventions to repair/update/reconfigure the rail control system, in a trusted way through the blockchain.

Finally, let us stress that FUSION is aligned with the current technology management trend of outsourcing the maintenance activities to third parties instead of relying on in-house maintenance divisions. Along this direction, the railway business use case allows us to show the advantages of adopting MDE and continuous assessment to evaluate the impact of adopting different predictive maintenance strategies on the reliability and availability of the railways infrastructure.

The rest of this article is organized as follows. Section II provides the needed background knowledge and position our work with respect to the state of the art. Section III details the FUSION platform, while our railway business case is presented in Section IV, along with preliminary results of the application of the continuous assessment approach. Section VI concludes this article.

II. BACKGROUND AND RELATED WORK

We present background notions and we position this article with respect to the related work in the IIoT domain by considering the pillar areas of our research: SDN, fog and edge computing, trust management, and continuous assessment, anticipated by a short introduction to blockchain.

A. Blockchain

A blockchain [8] is a distributed and immutable ledger that stores blocks of data containing transactions between nodes in a peer-to-peer network. It allows realizing a global consensus model despite Internet scale possible attacks/faults. A block contains batches of valid transactions inserted within the chain after being validated. In case of concurrent block additions by multiple nodes, a temporary fork can be produced, which is resolved afterward when the longer fork dominates and substitutes the other one. The integrity and immutability of the blocks is guaranteed by proper hash functions (generally a Merkle tree root hash) that allow us to have a block containing the hash of the previous block. To this aim, any modification of the data within the distributed ledger demands modifying all previous blocks, which is possible only with the consensus of the network majority. The way new blocks are validated defines the consensus model implemented by a blockchain platform, and one of the most known approaches is the Proof-of-Work in the Bitcoin blockchain [8]. To reach the consensus each peer, called *miner*, has to demonstrate to have completed a difficult cryptographic challenge by spending considerable computing resources, hence making impossible the manumission by malicious users. Blockchain is also used to implement smart contracts, i.e., applications that run in a distributed fashion on the blockchain to enforce and verify an agreement upon a given computation.

Blockchain has met enthusiastic adoption in many application domains differing from the financial one for which it was meant for, and IIoT does not make an exception [9]. The most natural use of blockchain within IoT (and IIoT in particular) is to provide decentralized security and privacy [10], by resulting in a low susceptibility to manipulation and forgery of data spread across the infrastructure. However, this is achieved at a high cost in terms of computing capacities/memory and energy consumption that the resource-constrained IIoT devices do not hold. Access control in IIoT is optimized by leveraging on blockchain for managing identity and authorization claims [11]. In [12], blockchain-based distributed cloud architecture based on SDN is proposed so as to cope with the demand of scalable communications. In this examples, blockchain is used to avoid the presence of a trusted intermediary for the security solutions, which can be designed in a distributed manner.

B. SDN in the IIoT

SDN is gaining more and more attention as a new model to overcome traditional issues of network management solutions, such as limited reconfigurability and complexity of managing traffic in a per-flow differentiated management [13]–[15]. The well-known main principle of SDN is the clear division between the *control plane* and the *data plane*. The former does the following:

- 1) achieves a logically centralized point of view of the network;
- 2) gathers application-level requirements;
- 3) makes control decisions based on the centralized point of view;

- 4) dynamically reconfigures nodes to ensure the achievement of targeted goals.

The latter is in charge of dispatching packets from sources to destinations, by transparently taking advantage of the control plane, which properly configures the mechanisms that rule how nodes should manage incoming/outgoing traffic.

SDN has emerged in the communication research and industrial fields of IIoT [3] primarily to manage switches of closed environments such as datacenters and department networks via the OpenFlow protocol [16]. More recently, Li *et al.* [17] focused on the adoption of the SDN paradigm in the context of IIoT to dispatch packets with different delay constraints in a per-flow tailored manner, by considering time deadlines, traffic load balances, and energy consumption. Similarly, Kaur *et al.* [18] adopted SDN to efficiently manage the interplay between edge and cloud environments by considering energy efficiency, bandwidth, and latency. In particular, this article proposes to differentially manage the huge amount of traffic between the edge and the cloud generated by the IIoT by selectively either activating or suspending traffic flows transmitted back and forth. In [19], authors exploited the SDN approach to increase the performance of a wireless sensor network (WSN) within an industrial environment. By adopting SDN, it is possible to facilitate deployment of personalized applications on-top-of WSN nodes, also reducing the energy consumption and load of some nodes while improving its overall reliability. Finally, Qiu *et al.* [20] originally proposed to adopt blockchain in an SDN-enabled IIoT scenario. The blockchain technology is primarily used to reach consensus among multiple SDN controllers. In particular, the blockchain represents a trusted third party useful to gather and synchronize information among distributed and independent SDN controllers. Similarly to [20], we propose to use the blockchain to validate distributed control decisions among SDN controllers. However, we further extend this concept, proposing the use of the blockchain also for the management of trust, as described next in this article.

C. Trust Management in the IIoT

Trust management [21] is a key aspect to consider when offering security and dealing with internal attacks (perpetrated by legitimate entities of an infrastructure by having a malicious behavior) within the context of federated architectures, such as the IIoT [22], [23].

There are two key problems in how implementing trust management, which severely influence its efficiency and effectiveness within the context of IIoT. On the one hand, despite of how to practically realize the overall process of trust estimation, a key concern is its robustness and tolerance to possible slander and/or false-praise attacks. Specifically, malicious adversaries may try to compromise the process of trust estimation by sending false reputation scores during their collection. It is possible that such false scores may be worse/better than the real ones with the intention of having an erroneous trust estimation, and the attack can be performed independently by the compromised reputation senders or even a group of attacking senders may cooperate in the attack. For this aim, when aggregating the reputation

scores, several means to detect and remove potentially false scores are used, such as a proper threshold [24] on the distance of each score from the average computed over the rest of the collected scores. However, it is reasonable to assume different formulations for the trust estimation [25] where a multicriteria characterization of trust is defined. With such a formulation it is challenging to apply a proper threshold scheme for the efficient detection and removal of false reputation scores. In [26], the Dempster–Shafer theory has been applied to deal with the uncertainty in the collected reputation scores, and to introduce the concept of Entropy [27] within the trust estimation process, a measure of divergence drawn from the information theory. However, these techniques are not completely secure and proper means to guarantee integrity and authenticity of the reputation scores are needed.

It is also worth noting that IIoT devices are characterized by limited computing resources and a nonrechargeable battery. Therefore, it is important to limit the number of exchanged messages (which represents the most power-consuming operation that such a kind of node may be involved in), or to do not use too complex cryptographic primitives.

To deal with such issues, in FUSION, we propose to use the blockchain technology to provide an eventual consistent view of the trust degree: any contacted node within the platform returns the same value of trust degree, obtained by querying the blockchain. To cope with the limited resources, the blockchain is deployed within the fog node, whose nodes are contacted by the IIoT devices.

D. Fog and Edge Computing for the IIoT

The integration of fog and edge computing in the last recent years enables the full convergence of heterogeneous wireless technologies. Furthermore, it seamlessly supports the delivery of a wide spectrum of different applications with highly challenging and very diverse requirements, such as in terms of latency, bandwidth, management, and so forth. In this area, there are two classes of proposals emerged in the last years. On the one hand, the platforms/solutions that start to be available under the name of fog computing (i.e., by Open Fog Consortium), have recently emerged as important enablers for IIoT scenarios [28]. On the other hand, the standardization efforts under the name of mobile edge computing (MEC) (i.e., by European Telecommunications Standards Institute—ETSI [29]) are currently under development to bring edge computing benefits into next generation 5G telco provider network deployments. Both proposals promote a new three-layer device-intermediate layer-cloud hierarchical architecture, which is recognized as very promising for several application domains. MEC stresses more the possibility to host computing/storage resources at network edges close to the targeted mobile devices, typically under the control of the telco provider. Fog computing, instead, focuses more on the composition of resources and services offered by all the local devices, which can usually interact thanks to the intermediation of a local proxy, in this context often called smart gateway (SG), and via direct point-to-point ad-hoc IIoT interconnections.

At the current stage, however, MEC and fog still face, each of them separately, some nonnegligible incompleteness and weaknesses. Starting with MEC, the number of employed edges is generally limited, since edges introduce additional costs of operation for supported services, such as deployment, maintenance, and configuration costs for telco operators. Moreover, MEC typically works in infrastructure mode, being unable to easily leverage the resources available in surrounding devices at runtime: once MEC edges are deployed, they are rarely and hardly redeployed in other positions (high cost of reconfigurations) and this might be highly inefficient, e.g., when service load conditions significantly change during provisioning, such as during specific time slots, maybe with daily, weekly, or yearly patterns. Focusing on fog, instead, although its more decentralized architecture (at least from a control/management perspective) makes it more flexible, at the same time it complicates its management and the possibility to leverage the monitored context (e.g., resource usage and availability) typically available in infrastructure-oriented MEC telco environments. In addition, fog use cases are tailored mainly for resource-poor devices and sensing scenarios, and so SGs are typically unable to host heavy computations, such as in the case of the execution of some core operations in the case of blockchain, such as mining.

To overcome such limitations, FUSION integrates the best of the two MEC and fog approaches by merging them into a unique, fully converged, and rich architecture that includes multiple classes of intermediate-level nodes with different hardware/software capabilities, as better explained in the following sections.

E. Continuous Assessment in the IIoT

IIoT assessment is fundamental to measure the *resilience* and *security* of the interested services. However, considering the scale, complexity, and evolutionary nature of the IIoT, the assessment of resilience and security properties represents a major technical challenge. In fact, sensors, actuators, hardware, software configurations, and services in the IIoT evolve through time, and give birth to a complex and evolutionary highly distributed CPS, composed of multiple loosely connected and heterogeneous parts [31].

Given the complexity of IIoT, assessment approaches cannot consider all possible evolutions of the system and its requirements. While most of the available assessment methods are based on the construction and evaluation of models representing a static view of the system, with predefined requirements and system structure, in the IIoT it is often difficult to define *a priori* risks, failure modes, and dependability and security requirements before deployment [32].

We advocate that mastering the evaluation of IIoT, maintaining the right level of detail, and at the same time accurately modeling all the interactions between system components require an approach that can operate also at runtime [33]. Assessment for understanding the system resilience and security properties should be performed continuously, on the basis of detected system evolution, and to provide feedback on the current status and consequent forecast on system behavior in the future (e.g.,

time to fail). However, performing runtime evaluations on the whole system is typically difficult or not feasible at all, either because it is too expensive and dangerous, or because obtained results may be scarcely representative of the actual system operation, due the high variability of its properties and of the environment [32], [33].

Performing runtime assessment requires three successive actions that are understanding evolutions of the system, describing the modified systems, and analyzing it. To such extent, the interplay between monitoring and fault management components to correlate at runtime observations, diagnosis outcomes, and decision/remediation actions in evolutionary systems have been largely explored in the past. For example, several research works have been done in the field of control theory and autonomic computing [34], as well as for monitoring quality of service (QoS) of systems and networks [35]. Approaches have spanned from analytical to experimental, e.g., runtime verification has been largely proposed to monitor the execution of programs and dynamically check if some properties are fulfilled during execution [5], [36], while several runtime testing frameworks have been proposed in various domains [37], [38]. However, the state of the art is still largely unable to answer appropriately the topic of continuous assessment of complex, heterogeneous, and evolutionary systems as the IIoT that we are targeting in this article. We believe that the effective approach consists in combining modeling and monitoring. In FUSION, we promote a dynamic model generation (and evaluation) process, capable to dynamically produce at run-time different models representing the current system state and conditions, and capable to feed the models' parameters with values coming from monitoring and experimental evaluation activities. The ultimate objective is to measure the resiliency and security properties of the current system configuration that will hold until the successive evolution.

III. FUSION PLATFORM

To achieve a tradeoff between flexibility and trust in the IIoT, we propose the combined use of blockchain, SDN, and cloud orchestration technologies at edge/fog levels. Our objective is to provide a fully distributed, scalable, resilient, secure, and verifiable architecture, named FUSION, for trusted flexible network management, configuration, and communication in the IIoT. The FUSION architecture aims to become a reference solution for the definition and assessment of global-wide blockchain-based management solutions for the IIoT.

The idea is to provide management and configuration flexibility by adopting FESNs, i.e., computing nodes acting as access nodes, SDN nodes, fog nodes, etc., depending on application needs. Network level management operations will then be performed adopting the SDN approach, to flexibly adapt the behavior of nodes in relation to the current requirements of the whole environment to support QoS. The use of blockchain technologies is envisioned to both perform decentralized trust management (to assure data integrity) and to allow secure and resilient management/configuration operations at SDN and FESN levels. Finally, the platform will be equipped with monitors to perform the continuous assessment of resiliency and security in

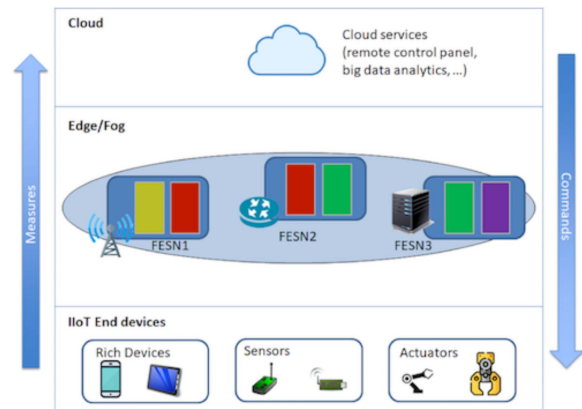


Fig. 2. FUSION high-level architecture.

the IIoT, exploring the joint use of simulative models derived from the architectural design and data collected at runtime from FESNs.

Fig. 2 outlines the high-level architecture of FUSION. FESNs have a central role and operate at edge/fog level. They can have different capabilities easily configurable depending on their role in the IIoT infrastructure. At the bottom there are IIoT end devices that often do not have enough capabilities to satisfy strict requirements on computation and response time; therefore, they delegate most functions to the upper layers. Finally, the cloud layer on top assists the intermediate edge/fog layer in supporting wide area network functionalities, e.g., permanent participation to the global blockchain.

FUSION exploits lightweight cloud and container technologies, such as OpenStack and Docker, to grant resource isolation and to simplify the deployment and fasten the de/activation of support functions and services at FESNs. Moreover, the platform exploits recent achievements in different areas that go from security to resiliency with blockchain, from Management and Orchestration (MANO) support to SDN for IIoT to take over all needed management issues [39], [40].

A. FUSION Edge Smart Nodes

Delving into finer details, the resulting FUSION architecture is flexible and composable enough in order to: 1) accommodate highly heterogeneous FESN and end-user IIoT devices; 2) support highly differentiated application scenarios, each one using the blockchain for slightly different purposes/application needs.

Toward that goal, each FESN node supports the following four main classes of functionalities that can be plugged at FESN nodes related to

- 1) participating to the blockchain network;
- 2) SDN management and deployment;
- 3) lightweight MDE monitoring;
- 4) advanced sensing activities such as aggregation/security/storage of IIoT data.

In particular, according to the node hardware capabilities and related supported functionalities, as shown in Fig. 3, we define two main classes of FESN: Level 0 FESN (i.e., L0FESN), and Level 1 FESN (i.e., L1FESN). Depending on the FESN class and the deployment context, we tailored the four main

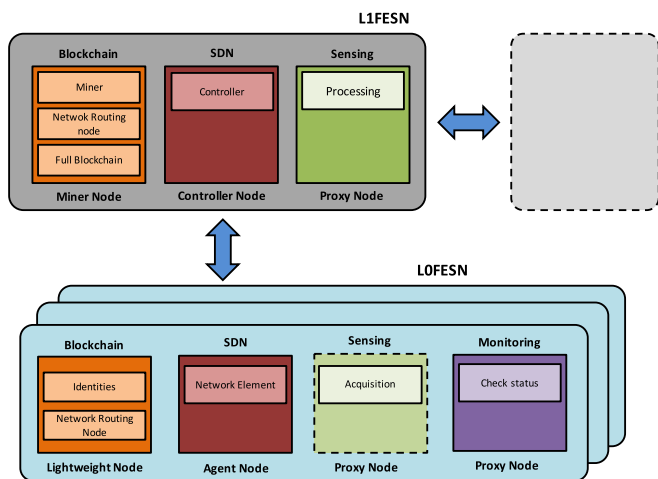


Fig. 3. FUSION L0FESN and L1FESN functionalities.

FESN functionalities and define specific roles through their composition; moreover, we organize them in a hierarchical architecture. At the lower layer, closer to the sensor layer, L0FESNs organize themselves in local networks, while at the upper layer L1FESNs act as local coordinators for L0FESNs under their management responsibility as well as representing local gateways. Then, L1FESNs, similarly to what happens for distributed SDN controllers, can be organized in more complex hierarchical deployments, such as for the sake of scalability.

With a closer view to details, L0FESN consists of edge/fog nodes with more limited capabilities. With regard to sensing, L0FESN typically works at a lower level by interacting with (nonsmart) sensors/actuators to dispatch data/commands thus realize basic *data acquisition* functions (see Fig. 3). Then, L0FESNs can interact in an ad-hoc mode with their peers to realize impromptu local networks, acting as *SDN* network elements; they also implement blockchain lightweight node functions, including identity and network routing services to be part of the blockchain overlay. Finally, L0FESN participates to continuous assessment activities, acting as local probes—monitors—able to check current status. At a higher level, L1FESNs add SDN controller capabilities, by realizing a distributed controller SDN plane. The higher resource availability at this level justifies also the hosting of advanced functions that are typically resource-hungry. That is the case of the sensing aggregation and processing as well as the blockchain mining functions, thus acting as full-fledged blockchain nodes.

Finally, to enable function composability, FESN at both levels offers adequate management support as shown by the internal architecture in Fig. 4. First of all, FESNs provide abstracted APIs transformed into composable FUSION functions, by adopting an approach similar to network function virtualization (NFV [41]). We adopt Docker as the container solution for our implementation, as shown in the figure. Moreover, each FESN features a service orchestrator to take over the MANO of the whole infrastructure by also enabling the dynamic (re-)composability of those virtual network functions (VNFs) as needed at runtime. MANO is the ETSI-defined framework for the MANO of all resources. This includes computing, networking, storage, and

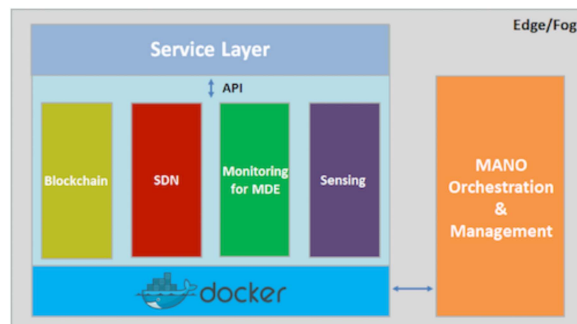


Fig. 4. Internals of an FESN.

virtual machine resources. The main focus of MANO is to allow flexible on-boarding and sidestep the chaos that can be associated with rapid spin up of network components. MANO is composed by the following three main functional components:

- 1) NFV Orchestrator, responsible for on-boarding of new network services and VNF packages;
- 2) NFV Manager, oversees lifecycle management of NFV instances;
- 3) virtualized infrastructure manager, controls and manages the NFV compute, storage, and network resources.

We use a particular solution based on MANO framework called OpenBaton; OpenBaton extends the existing standard specification of ETSI MANO to properly manage also MEC applications and to use container deployment tools.

B. Blockchain-Based SDN in Edge Networks

As anticipated in the previous section, regular FESN nodes, namely L0FESNs, interact with sensors/actuators to dispatch data/commands. In addition, they cooperate one each other to create a multihop (and eventually even multipath) edge networks to distribute packets at multihop distance. Moreover, for each edge network there is one L1FESN behaving as edge SDN controller, in charge of supporting the proper management of resources both at the network and at the application level.

On the one hand, at the network layer L1FESNs gather information about the state of the edge network, enforce traffic engineering policies, and interact with other SDN controllers outside the edge network, e.g., cloud SDN controllers, to take proper management decisions considering traffic flows from the cloud to the edge and viceversa. For instance, by specifically considering the interaction with remote cloud SDN controllers and local L0FESNs, L1FESN nodes:

- 1) prioritize traffic flows within an edge network in relation to its importance, e.g., by temporarily freezing sensed data dispatching to ensure the prompt delivery of reconfiguration messages;
- 2) tune the quality of traffic flows, e.g., by selectively dropping (part of) low-priority traffic flows if the current bandwidth is limited;
- 3) reroute traffic flows toward the destination, considering the current location, direction, and speed of eventually moving edge networks.

On the other hand, at the application layer L1FESN nodes:

- 1) receive application-dependant control commands from administrators, e.g., technicians interacting with sensors either locally or remotely from the cloud, and dispatch control/configuration messages to L0FESN nodes;
- 2) efficiently manage sensed data to be delivered to remote locations, e.g., by creating tree-based overlay networks to minimize the latency of data delivery along edge nodes and also supporting the dynamic deployment/configuration of preprocessing mechanisms to reduce the amount of data delivered to the cloud.

To support the flexible and dynamic QoS management we adopt the multilayer advanced networking environment (Multi-LANE) solution [42]. Multi-LANE dynamically selects and exploits (even at the same time) different routing strategies and mechanisms suitable for applications with heterogeneous features and requirements. In particular, based on its centralized point of view our Multi-LANE SDN controller determines the most suitable path and configures the proper forwarding mechanism.

Let us note that the adoption of the SDN actually improves network flexibility and efficiency, since the (per edge network) centralized L1FESN can maximize a edge network performance considering the current state of the network as well as it can easily reconfigure the network in case there is the need to cope with new application needs and requirements. However, it strictly relies on the assumption that there is always an active and, most relevant, trusted L1FESN in charge of managing the network. In case there is no L1FESN (since it failed) or it is not trusted (since it has been compromised) L0FESNs cannot operate in a proper manner, eventually compromising the regular behavior of the whole edge network.

To overcome this limit and provide a resilient solution, we propose to exploit the blockchain technology to securely store management commands and sensed data. In particular, the adoption of blockchain in an SDN-based edge network allows us to improve resiliency since the network can still work properly even if it is either disconnected or controller-less. In fact, the blockchain allows us to support secure service provisioning also in case the network becomes partitioned/disconnected or server-less.

In case of partitioned/disconnected edge network, i.e., (part of) it does not have Internet connectivity or the bandwidth toward the Internet is limited, important events can be still securely recorded, since saved in the blockchain. Then, once the edge network is online again, past remote events can be reported even if in the meanwhile the reporting node has leaved/failed since sensed data are securely stored among L0FESNs in a distributed manner based on the blockchain.

In case of server-less edge network, i.e., the L1FESN failed, configurations/commands (eventually sent by different admins) are stored by L0FESNs in the edge blockchain. In this manner, even in case the L1FESN fails newly joined L0FESNs can get configurations from the blockchain, i.e., in case a new node is added while the network is controller-less, the blockchain can provide secure and trusted configuration information to the new FESN. Then, when a new L1FESN is activated (eventually based on L1FESN auto-election mechanisms, out-of-the-scope of this

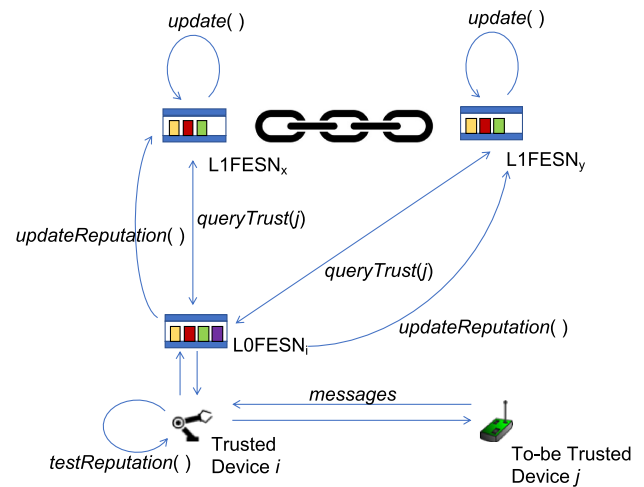


Fig. 5. FESN-based and blockchain-enabled trust management within the context of IIoT.

article), the new L1FESN securely receives the previous network configuration from the blockchain, e.g., flow ids, reroute rules, and traffic engineering policies. Note that everything works fine even if the new and previous L1FESNs are different, since the state (and also the history) of the network is distributively and securely saved edge-side in the blockchain, not in a centralized controller node.

C. Trust Management With Blockchain in FUSION

On the one hand, due to the presence of mobile nodes and the large number of devices within the envisioned IIoT, it is not practical to have a centralized approach to trust management, where a single node collects the reputation scores related to all the entities to be assessed and aggregates such scores. On the other hand, to meet a high level of energy efficiency, the traditional distributed approach of IIoT devices exchanging messages and computing trust degrees is not viable too, due to the high number of exchanged messages. In FUSION, we propose a tradeoff between the centralized and distributed solutions, by having FESNs responsible to perform the trust management in a federated manner, as depicted in Fig. 5. Specifically, when a given trusted IIoT device (e.g., the actuator i in the figure) interacts with another to be trusted device (e.g., the sensor j in the figure), for which the trust degree is unknown or too old, it sends a request through its L0FESN $_i$ toward all the reachable L1FESNs, e.g., L1FESN $_x$ and L1FESN $_y$, which reply by returning the trust value.

The trusted IIoT node continuously monitors the behavior of the other node, and it can update the trust value (by sending the new value to L1FESNs). The consistency of the information hold by FESNs is maintained thanks to the blockchain, so that trust values cannot be unilaterally altered by a malicious end device or L0FESN: all the correct L1FESNs always return the same trust value. This allows an IIoT device (or L0FESN on its behalf) to detect a malicious L1FESN, as the one returning a different trust value than the majority of the contacted ones.

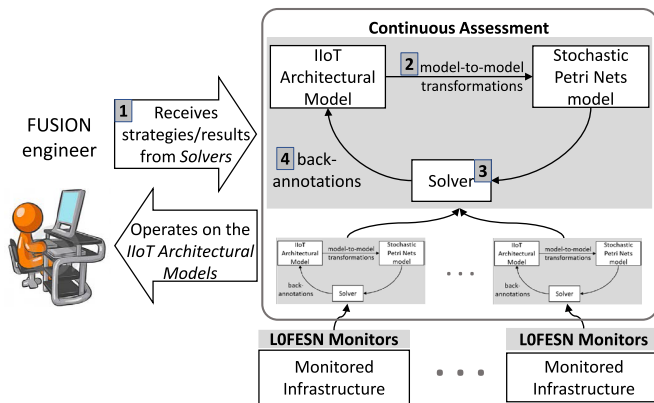


Fig. 6. Workflow and actors for continuous assessment.

Similar approaches are used throughout the FUSION platform for other purposes, e.g., for maintenance/configuration data management and/or for distributed SDN controllers status update, as seen in the previous paragraph. The specific blockchain platform used for prototyping the approach presented in this section is Hyperledger Fabric [55].

D. Continuous Assessment Methodology

The overarching workflow of the FUSION continuous assessment approach is presented in Fig. 6. The workflow is supported and integrated within the CHES tool [43], a framework for developing an industrial-quality MDE infrastructure that permits high-integrity embedded systems to be assembled in a component-based fashion.

The starting point of the workflow (box 1 in Fig. 6) is the definition of an architectural model (IIoT architectural model) based on the current state of the system, which is performed by the FUSION engineers/technicians adopting the CHES tool and using a language, based on the unified modelling language (UML), called CHES-ML (see [43]–[45]) for modeling the IIoT architecture and its key dependability aspects. In particular, the model specifies the architectural components (hw/sw), their relations (e.g., deployment relation), the attributes characterising their dependability and security properties, and the metrics to be analyzed (e.g., reliability of a component or of a part of the system). The resulting IIoT architectural model can then be automatically analysed by the CHES tool, which allows 1) to automatically apply a model-to-model transformation to derive a stochastic Petri nets (SPN) analysis model from the IIoT architectural model (box 2 in Fig. 6), and 2) to automatically solve the analysis model for computing the required reliability and security metrics (box 3 in Fig. 6).

To support an iterative and incremental development process, analysis results are then used to enrich the initial architectural model from which the analysis has been triggered, i.e., automatically updating the IIoT architectural model through the setting of some of its model parameters (back-annotations, box 4 in Fig. 6). Such automatic updates will allow considering in the next iteration of the analysis the most updated results representing the metrics of interest. For instance, the probability of an omission

failure of a specific IIoT component, computed in an iteration of the FUSION continuous assessment framework, could be used in the following iteration for assessing the probability of catastrophic failure of a part of the IIoT that includes such component. Besides allowing the automatic model update, analysis results are also dispatched to the FUSION engineers/technicians who can further check and analyse them, and then manually implement further modifications to the IIoT architectural model that cannot be automatically implemented through the back-annotation (e.g., integrating an additional architectural component, or changing some fault tolerant mechanisms). These manual modifications need to be considered in the development process due to the evolutionary nature and complexity of the IIoT and the impossibility to consider in advance all the possible future scenarios.

Finally, the computed results are also propagated to the other L1FESN nodes for taking the appropriate system reconfiguration decisions and dispatching the control/configuration messages to LOFESN nodes, e.g., triggering some preventive maintenance actions on some IIoT components.

The entire workflow can be executed hierarchically. Each hierarchy level provides the system knowledge to its upper level, used to build the IIoT architectural model. Information provided by the LOFESN nodes, data from solvers at lower levels, and results from stochastic analysis provide back-annotations to improve the CHES-ML architectural model. The application of back-annotations prescriptions to the IIoT architectural model requires to iterate the workflow to update the model and perform a new stochastic analysis. Iterations are expected to be activated: 1) by the FUSION engineers/technician, who manually modifies the CHES-ML model; 2) automatically, triggered by the FUSION itself, when the information received from a lower level prescribes different parameters settings (for example, with a more accurate setting of a model parameter as a failure rate) instead of manual modifications of the IIoT architecture.

In the rest of this section, we further discuss some of the key elements of the FUSION Continuous assessment methodology, concerning the IIoT architectural model construction and the generation of the SPN model.

Dependability and security concerns in IIoT: CHES-ML includes specific UML extensions for describing the dependability and security aspects of an IIoT infrastructure. Such extensions are grouped into three abstract levels, whose concepts reflect the classical taxonomy that classifies dependability and security aspects as threats (to dependability and security), means (to attain dependability and security), and attributes (of dependability and security) [46]. A fourth level structure takes into account the structure of the system, by identifying basic architectural components of the system and their relations. The key dependability and security aspects that can be modeled with CHES-ML are as follows.

- 1) Threats and propagation: definition of the threats affecting system components and their propagation paths.
- 2) Risk: specification of the risks and related safety properties.
- 3) Fault tolerance: definition of the fault tolerant structures and mechanisms.

- 4) Risk mitigation: definition of the means for the mitigation of risk deriving from failures identified in the lower layer.
- 5) Maintenance: definition of the maintenance activities and policies.
- 6) Requirements: specification of dependability and security requirements.
- 7) Metrics: specification of the metrics to be evaluated by the stochastic analysis technique.

To attach dependability and security information to system components, we define a specific set of stereotypes provided by CHESS-ML, which describe typical categories of components and allow their dependability and security properties to be specified based on a simple set of attributes. Such stereotypes provide templates for classes of components common in dependability analysis, in order to save the modeler from providing excessive redundant information [44].

Automated generation of stochastic state-based models: In accordance with MDE principles, the analysis models are automatically derived from the high-level model describing the system's architecture. We selected SPN with general probability distributions as the analysis formalism. This choice is mainly due to the intent of supporting: 1) nonexponential occurrence of faults (e.g., for mechanical components), and 2) periodic maintenance schedules. The model is then evaluated using a discrete-event simulation integrated in the MDE methodology [45].

IV. BUSINESS CASE

To demonstrate the feasibility and the effectiveness of FUSION, in the following, we apply it to the railway application scenario. In particular, we are interested in supporting the whole (complex) process of keeping track of the management tasks required to repair/update/reconfigure the rail control system from both software (new versions of software, new components, configurations, etc.) and hardware (new sensors, new gateways, IoT devices, etc.) perspectives. We chose this specific case because it is very rich in terms of complexity and there are also some regulatory and ethical constraints to comply with. In addition, some of the authors are already involved in related projects, e.g., funded by the European Commission within the context of Shift2Rail [48], a European initiative to seek focused research and innovation and market-driven solutions by accelerating the integration of new and advanced technologies into innovative rail product solutions.

Focusing on technical aspects, first of all this scenario calls for a secure support able to register in a noncorruptible way all needed on-the-field management and reconfiguration actions to track any malicious misconfiguration without having to trust on a third-party (typically centralized) entity. This kind of support is the one offered by blockchain. Moreover, the railway system demands the timely collection of complete and up-to-date information on the behavior of its components to be able to maximize performances, perform early planning of maintenance activities, and improve return-on-investment. This is of paramount importance to keep and increase the competitiveness of such a complex infrastructure, to respond to Europe's need for sustainable and safe mode of transport. The available budgets to maintain and

renew the railway infrastructure have been reduced throughout Europe, resulting in a need to keep a high level of quality at lower cost. Leaving the traditional proactive maintenance approach and adopting a predictive one allows infrastructure managers to increase the infrastructure efficiency by predicting when maintenance should be performed [49], [50]. In its turn, this enables cost savings over routine or time-based preventive maintenance because tasks are performed only when needed. Moreover, we are witnessing a slow, but certain, radical change where more and more railway operators and managers outsource maintenance activities of rolling stock and infrastructure to third parties instead of relying on monopolistic in-house maintenance divisions, so as to cut costs and increase efficiency gains [51]. With the increasing degree of liberalization, the number of actors in the railway sector has multiplied, competing among each others. There is the urge of tracking the conducted maintenance actions (by registering what has been done and by who) so that in case of errors it is possible to trace back the culprit and prosecute him/her. Furthermore, all contracts with third party maintainers are based on key performance indicators, which must be precisely and timely assessed, so that such maintainers can receive the agreed fee only if the targets on their contracts are actually met.

A large amount of diagnostic data from the signalling and telecom systems is needed to achieve worthwhile predictions. However, this is currently a challenge, since there are several problems to be handled, from the presence of multiple proprietary interfaces for data gathering that are not interoperable among themselves, to a scarce interconnection among information and communication technology (ICT) systems deployed along the infrastructure. Specifically, the lower part of Fig. 7 presents a generic piece of a railway infrastructure, with multiple signalling entities, such as switches, lights, track circuits for train detection or balise. All these elements within a certain geographical area exchange data (by means of a proprietary protocol and data model/format) with an interlocking system deployed at the main station of the area. The diagnostic data are presented to a human operator, and scarcely exchanged with similar neighbor systems, unless their manufacturer is the same. Current research efforts within the Shift2Rail framework focus on the interoperability of such systems, so as to feed a predictive model running within the cloud. Proper software to enhance the system interoperability are under design and will be deployed between the interlocking systems and the cloud, or even within the racks of such systems in a near future without violating the safety cases and their certification. Such software can be considered as L0FESN nodes within the FUSION vision, and may not directly interact with the cloud, due to scalability reasons, but may leverage on a set of L1FESN nodes to federate neighbor interlocking systems, preprocess the incoming data, and return to the cloud the obtained data so as to ease the workload applied to the software running within the cloud. The multiple maintenance companies may have access to those fog nodes so as to obtain diagnostic data before an action, report the result of a performed operation, or even claim the fee of the conducted maintenance.

To support this business case, we adopt FUSION. First of all, the FUSION blockchain-based registry has a pivotal role as a

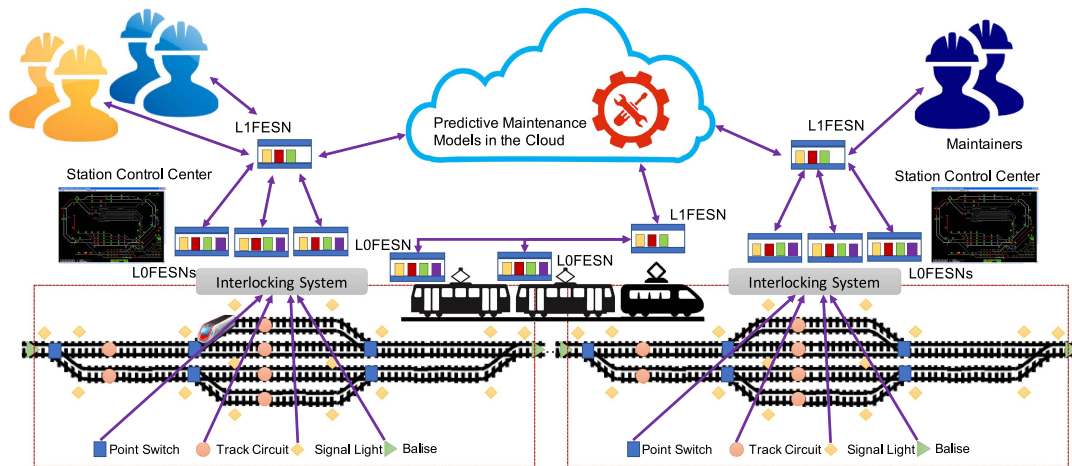


Fig. 7. Railway infrastructure and the FUSION infrastructure.

tool to track, in a distributed, transparent, and shared view, all the actions (installation of new devices, on-the-field operations, reconfigurations at various software stack levels, etc.) in a fully decentralized way, by tracking the role and actions of all the main actors in the overall logistics/supply chain. This is particularly crucial in the current scenario of the maintenance of rolling stock and infrastructure carried out by a number of third parties instead of in-house maintenance divisions of the state railways. Even if the maintenance actions can be outsourced, responsibility cannot, so infrastructure managers are responsible not only for their own actions, but also for those of their contractors. The blockchain-based registry envisioned by FUSION supports a comprehensive inventory of all the maintenance actions carried out by third parties with whom the railway manager has a relationship, monetize each intervention, trace back a specific action and maintenance team in case of an issue/accident.

To this purpose, we adopt the SDN approach to more easily manage the incoming/outgoing traffic from/to the cloud (typically carrying reconfiguration commands and sensed data respectively) as well as the traffic within each train (also carrying information about the current state of FESN nodes).

In particular, we adopt a multilayered federated approach characterized by different SDN domains. An LOFESN node is deployed on each train wagon, with the main purpose of directly interacting with nearby sensors and actuators. LOFESNs in different wagons of the same train interact one each other to dispatch packets among them. In addition, an L1FESN node is deployed on each train, acting as edge/fog SDN controller and remotely interacting with cloud SDN controllers. Moreover, each station is composed of multiple LOFESN nodes, e.g., interacting with point switches, track circuits, signal lights, and balise, and a logically centralized L1FESN node, thus also providing edge/fog SDN controllers capabilities by monitoring/managing the local station network and interacting with cloud ones.

L1FESN nodes on trains and stations interact with cloud SDN controllers to properly reroute traffic flows and manage their QoS. On the one hand, cloud SDN controllers exploit information about train direction and speed to reroute traffic flows toward the station the train is actually approaching, thus taking

advantage of the large bandwidth of train-to-station connectivity. On the other hand, in case the train is not approaching any station the traffic flow is rerouted toward the traversing rail line, typically characterized by limited bandwidth. For this reason, in this case edge/fog and cloud SDN controllers adopt traffic engineering techniques to dynamically manage traffic flows in relation to their priority levels, e.g., by temporarily delaying (or even dropping) the dispatching of low-priority sensed data in favor to the prompt dispatching of control/reconfiguration messages.

This scenario can greatly benefit from the adoption of the blockchain technology to securely store important sensed data and control messages, e.g., by allowing to greatly improve trust and resiliency within the train network. First of all, in case the train is temporarily disconnected from the Internet, sensed data can be cached by LOFESN nodes in a secure and distributed fashion, then sent to the cloud even if the LOFESN node that gathered data from sensors is not available anymore. Second, newly joined LOFESN nodes related to new wagons can securely retrieve previous configuration commands from other LOFESN nodes without any direct interaction with cloud SDN controllers, even if the edge one fails. For instance, LOFESN nodes can retrieve information about how to manage different traffic flows (i.e., their priority levels and if low-priority ones should be either delayed or dropped) only interacting with LOFESN nodes deployed in other wagons. Finally, every reconfiguration command performed by local technicians (e.g., manually and directly interacting with sensors and nodes) is tracked and securely stored, also ensuring nonrepudiation.

A. Continuous Assessment in the Considered Scenarios

In this section, we prove the feasibility of the continuous assessment methodology proposed in Section III-D considering its application to our business case. The monitored system is a portion of a railway infrastructure consisting of three balises, two signal lights, two point switches, and one track circuit, whose failure distributions follow a Weibull random variable. The goal is to assess the reliability and availability of the system

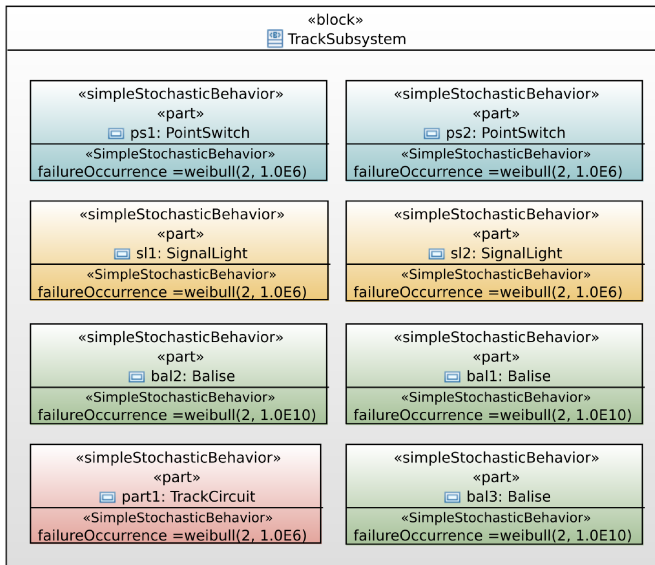


Fig. 8. CHES-ML model for the railway infrastructure.

considering the impact of adopting different preventive and corrective maintenance strategies applied to different infrastructural components, so to guide the FUSION technician in taking proper maintenance actions.

Fig. 8 depicts the CHES-ML model corresponding to the monitored infrastructure.

There are the following four types of elements in the system model, each one representing a kind of component: balise, SignalLight, TrackCircuit, and PointSwitch. Instances of these elements are defined as “part” elements, in accordance with the targeted infrastructure: three instances of the balise block, two instances of the SignalLight block, one for the TrackCircuit, and two for the PointSwitch. The failure distribution of each instance is defined as a Weibull with different shape and scale parameters (reported in 8), which can be continuously updated by FESN monitors. The different maintenance strategies are defined as part of an activity diagram.

We considered three scenarios, each one considering different maintenance strategies.

1) *Scenario 1*: In the first scenario, we define a maintenance activity that consists of periodically repairing every 5000 h both block instances ps1 and ps2, the two point switches in the infrastructure. Each repair will take half an hour to be completed (duration is deterministic equal to 0.5 h). The objective is to analyze the system reliability at time t , which is defined as the probability that all the components will be properly working (i.e., will not fail) throughout the interval $[0, t]$.

Starting from the CHES-ML architectural model, a sequence of automatic model-to-model transformations leads to the generation of an SPN model represented using the Petri net markup language (PNML), a proposal for a Petri net interchange format based on XML that is under development as an ISO/IEC standard, which is then further transformed in a “Deem Input file” for being evaluated using the discrete-event simulation of DEEM [47], a tool for the dependability modeling and evaluation of systems, based on deterministic and SPN and on Markov

regenerative processes. The “Deem Input file” is essentially a text file composed of different sections, containing (in the following order):

- 1) the definition of variables to be used in the study definition;
- 2) the studies to be performed on the model, i.e., the combination of different values for the variables specified earlier;
- 3) the list of places;
- 4) the list of transitions;
- 5) the list of arcs;
- 6) the list of measures of interest to be evaluated.

The results computed by the DEEM simulator are then stored in the Deem Results File, which contains the set of evaluated metrics, along with their mean, the confidence interval, and the number of samples (runs) on which they have been computed. More details on the transformation tool-chain can be found in [45]. We selected DEEM for the model analysis for a number or reasons:

- 1) it is open source, so we have full control of its required input (including the Deem Input file) and of its internal behavior;
- 2) it is a simulator that supports the analysis of SPN with general probability distributions, which is exactly the analysis model automatically generated from the IIoT architectural model;
- 3) it supports the analysis of the dependability (reliability, availability, safety) and security metrics we are interested in;
- 4) it provides all the key features common to all the discrete-event simulators, like the capability to represent the system state, to keep track of the simulation time, to maintain the list of the simulation events, to generate pseudorandom numbers, to keep track of system’s statistics, to define the ending conditions of the simulation.

Fig. 9 shows the SPN model with general timing distributions corresponding to the generated PNML model.

In the initial marking, places SysFailed and PSRepairing have no tokens while all other places contain one token. Each instance of the architectural components (track circuit, point switch, signal light, balise) corresponds to one input place. The failure of each instance is represented using a timed activity with Weibull distribution (failureOccurrence attribute of the block instances of Fig. 9), which produces a system failure (place SysFailed) when it completes. Activation and reactivation predicates are defined for both the activities ps1Fail and ps2Fail, for modeling the effect of the repair (substitution) of point switches. Specifically, when a token is added to the place Resample the firing delays of the two failure distributions of ps1Fail and ps2Fail are resampled according to the Weibull distribution (if not already failed), thus representing the substitution of the two old point switches with new ones. The frequency of the maintenance activity execution on the two point switches is represented by the deterministic transition RepairPS firing with a delay of 5000 h. When this transition completes, a token is added to PSRepairing representing the start of repair activities. The RepairPSDuration transition completes in a deterministic time equal to 30 min, thus triggering the resampling of the transitions ps1Fail and ps2Fail as previously detailed. Then, the completion of the instantaneous

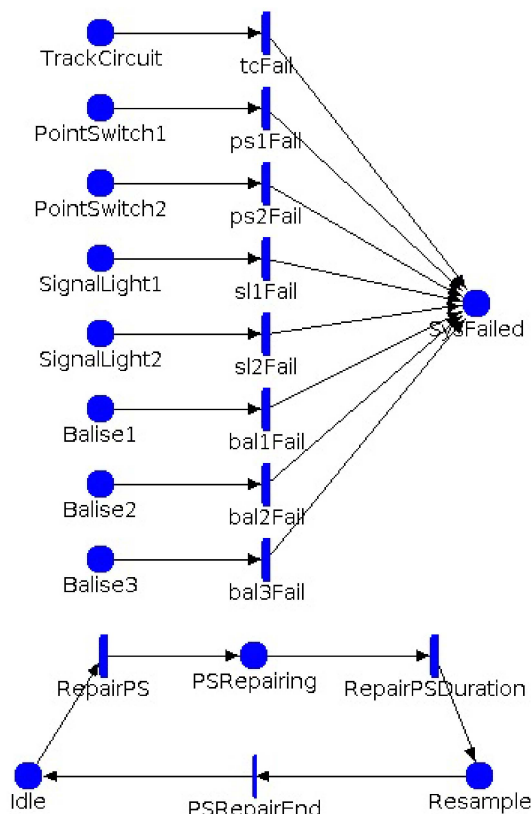


Fig. 9. Generated SPN model.

transition PSRepairEnd adds one token to the Idle place to enable the next repair activity.

Following the transformation process defined in [45], the SPN of Fig. 9 is then automatically solved using the discrete-event simulation provided by DEEM. The simulation model represents the initial state of the system and its changes over time: it takes as input the random variables corresponding to the firing delays defined in the SPN, and allows us to numerically compute the defined metrics of interest. In a simulation model, the metrics are not analytically derived from probability distributions, but rather as averages over replications that is different runs of the model. Confidence intervals are also defined for assessing the quality of the produced output. For each study presented in this section, we executed a minimum of 1 million simulation runs and we set the relative confidence interval to 0.1 and the confidence level to 0.95, with a maximum of 10 million runs. This means that the stopping criteria will be satisfied when the confidence interval is within 10% of the mean estimate in 95% of the times or the number of simulation runs is equal to 10 million.

Fig. 10(a) shows the results that will be back-annotated in the CHES-ML model and provided to the FUSION technician to support the maintenance decision.

The three plots in the figure show the system reliability at varying of time considering three preventive maintenance strategies, each one repairing the two point switches with a different period of time: every 5000h, every 10000h, and every 15 000 h. Other sensitivity analysis could be carried out at varying of other model parameters, like the scale parameter of

the Weibull failure distribution and the probability of successful maintenance execution, which may also be updated by FESN monitors.

2) *Scenario 2:* Let us now suppose that at time $t = 14\,000$ h several maintenance activities take place that consist of substituting all the components in the scenario with the exception of the three balises. The failure distributions of the new components still follow a Weibull distribution but with different scale parameters (changing from 1.0^{-6} to 7.5^{-7}). The FESN monitors will then trigger a modification to the CHES-ML model that consists of updating the scale parameters of the failure distributions of the maintained components, and a new iteration of the continuous assessment framework will take place. The new reliability results are presented in Fig. 10(b) showing the effects of the maintenance action at time $t = 14\,000$ hours on the system reliability.

3) *Scenario 3:* In this last scenario, the objective is to assess the system availability in a 1 year interval (computed as 360 days), which is defined as the fraction of time the system is available (not failed) in the considered time interval. For instance, an availability of $9.999E-1$ means that the system is not available for about 50 min in a year. We extend the configuration of Scenario 1 considering that each infrastructural components, once failed, can be repaired/substituted in a time uniformly distributed with mean repairTime in the interval $[\text{repairTime} \times 0.9, \text{repairTime} \times 1.1]$, thus representing the application of *corrective* maintenance actions to each failed components.

In Fig. 10(c), we show the impact of a combination of different maintenance strategies on the system availability for the next ten years of the system’s lifetime. In particular, we consider the possibility to apply the following:

- 1) only corrective maintenance actions, with different mean repair times (repairTime = 4 h, 1 week and 1 month—computed as 30 days);
- 2) both preventive and corrective maintenance actions, combining the previous corrective maintenance strategies with a preventive maintenance that substitutes the two point switches every 15 000 h (as in Scenario 1).

The results allow us to get the following insights on the system.

- 1) If the components can be repaired in a few hours (repairTime = 4 h), the benefit in applying both the preventive and corrective maintenance actions [plot “Prev. and Corr. (Rep. Time = 4 h)”] is quite limited with respect to applying corrective maintenance only [plot “Corr. only (Rep. Time = 4 h)”]. Actually, these two plots are overlapped in the figure since they differ of about $4E-6$ (about 2 min) in the worst case (for year 10).
- 2) While increasing the repair times, the benefits in combining preventive and reactive maintenance becomes much more significant. For instance, with repairTime = 1 week the availability improvement for year 10 is of about 35 min.

Such results can be used to support FUSION technicians’ decisions in selecting/updating the maintenance strategies to be applied depending on the current state of the system and

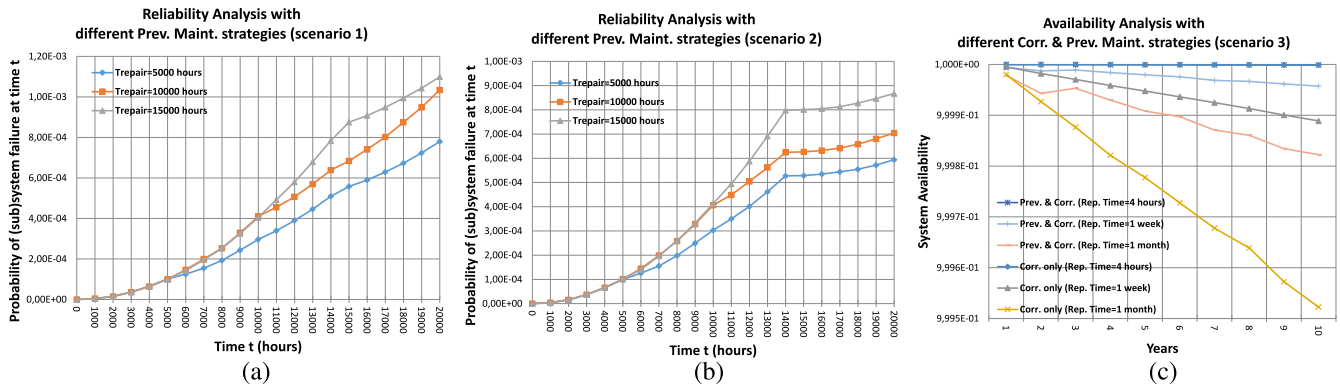


Fig. 10. Reliability and availability analysis for three scenarios.

on the reliability/availability levels to be guaranteed for the system.

V. MANAGERIAL IMPLICATIONS

From the managerial perspective, the introduction of the FUSION platform leads to several practical implications in real IIoT systems in terms of decentralization, remote management, and preventive management.

First, the adoption of blockchain allows us to decentralize trust information, assuring integrity, and allowing certified and trusted management operations. This in turn enables to trustfully outsource maintenance activities on actual industrial plants and critical infrastructures to third parties. In fact, repair/update/reconfigure interventions will be conveniently and transparently tracked and certified (and then billed) on the blockchain. Organizations can, thus, reduce the costs due to in-house maintenance divisions, moving toward a more convenient pay-per-use model.

Second, the use of the SDN paradigm (with reconfiguration actions treated as trusted transactions over the blockchain) allows us to remotely manage devices as virtual nodes, named FESNs, and optimize network and nodes resources based on application needs. In this manner, there is no need to physically reach the device that could be expensive and dangerous in realistic scenarios, as depicted in our railways business case.

Third, depending on the selected metrics continuous assessment may aim at cost optimization, reliability improvement, security improvement, etc. We showed that the adoption of MDE and continuous assessment built-in in FESNs helps to compare different preventive and corrective maintenance strategies on the physical plant, considering their impact on the offered system reliability and availability levels. The notable benefit is that FUSION adoption makes it possible to avoid potentially useless (and expensive) interventions, while focusing on only effective ones.

In addition to the individual features offered, the three elements of blockchain, SDN, and continuous assessment offer the most of their management engineering potential when they are considered in a coordinated fashion. In fact, these three enabling technologies offer to the management engineer the capability to implement a trusted control-loop process, which includes technological supports to monitoring, analysis,

planning, and definition of improvements or response strategies. The engineer can rely on continuous assessment to study novel configurations that can improve its own target metrics such as costs-effectiveness, reliability improvements, security, or that can enable rapid response. The SDN allows acquiring the data that make possible the continuous data acquisition, data processing and response. On top of this, the trusted decentralization of services and management offered by blockchain overcomes the bottlenecks problems of centralized decision in management. Overall that grants the capability of performing rapid data analysis and autonomous reconfiguration.

Hence, overall, FUSION enables several managerial advantages by easing the management of complex IIoT solutions by making the configuration and interventions traceable by paving the way to their trusted certification.

VI. CONCLUSION

In this article, we presented FUSION, a software platform born with the aim to show the advantages to combine the blockchain with SDN and container-based orchestration for the trusted management of devices in the IIoT. The platform was centered around the notion of FESNs as computing nodes for trusted IIoTs that can be flexibly reconfigured to act as end devices (L0FESN) or fog/edge nodes (L1FESN). We envisioned the use of the blockchain to both manage the trust information, in a federated way, and to securely manage FESN management in terms of reconfiguration. The presented railways business case has shown how the platform can simplify and make more trusted the intervention of technicians on equipment connected to edge devices, even when those technicians do not trust one another. This is also coupled with continuous assessment, which can be used to plan preventive maintenance, on the basis of tradeoffs between costs (e.g., how often to repair) and reliability, also taking into account possible changes that can happen during the life of the system. Again, monitoring information, collected by the trusted nodes of the system, is securely stored by FESNs using the blockchain.

REFERENCES

- [1] S. Jeschke *et al.* (eds.), *Industrial Internet of Things, Springer Series in Wireless Technology*, Oct. 2016. pp. 3–19, doi: [10.1007/978-3-319-42559-7](https://doi.org/10.1007/978-3-319-42559-7).

- [2] X. Li, D. Li, J. Wan, A. Vasilakos, C. Lai, and S. Wang, "Review of industrial wireless networks in the context of industry 4.0," *Wireless Netw.*, vol. 23, pp. 23–41, 2017, doi: [10.1007/s11276-015-1133-7](https://doi.org/10.1007/s11276-015-1133-7).
- [3] J. Wan *et al.*, "Software-defined industrial Internet of Things in the context of industry 4.0," *IEEE Sensors J.*, vol. 16, no. 20, pp. 7373–7380, Oct. 15, 2016.
- [4] R. Langner, "Stuxnet: Dissecting a cyberwarfare weapon," *IEEE Security Privacy*, vol. 9, no. 3, pp. 49–51, May/June 2011, doi: [10.1109/MSP.2011.67](https://doi.org/10.1109/MSP.2011.67).
- [5] R. Calinescu, C. Ghezzi, M. Kwiatkowska, and R. Mirandola, "Self-adaptive software needs quantitative verification at runtime," *Commun. ACM*, no. 9, vol. 55, pp. 69–77, 2012.
- [6] R. de Lemos *et al.*, "Software engineering for self-adaptive systems: Research challenges in the provision of assurances," in *Software Engineering Self-Adaptive System III. Assurances*. Cham, Switzerland: Springer, 2017, pp. 3–30.
- [7] M. A. Khan and K. Salah, "IoT security: Review, blockchain solutions, and open challenges," *Future Gener. Comput. Syst.*, vol. 82, pp. 395–411, Nov. 2017, doi: [10.1016/j.future.2017.11.022](https://doi.org/10.1016/j.future.2017.11.022).
- [8] A. Kaushik, A. Choudhary, C. Ektare, D. Thomas, and S. Akram, "Blockchain—Literature survey," in *Proc. 2nd IEEE Int. Conf. Recent Trends Electron., Inf. Commun. Technol.*, 2017, pp. 2145–2148.
- [9] N. Kshetri, "Can blockchain strengthen the Internet of Things?" *IT Professional*, vol. 19, no. 4, pp. 68–72, 2017.
- [10] M. A. Khan and K. Salah, "IoT security: Review, blockchain solutions, and open challenges," *Future Gener. Comput. Syst.*, vol. 82 pp. 395–411, 2018.
- [11] O. Novo, "Blockchain meets IoT: An architecture for scalable access management in IoT," *IEEE Internet Things J.*, vol. 5, no. 2, pp. 1184–1195, Apr. 2018.
- [12] P. K. Sharma, M. Chen, and J. H. Park, "A software defined fog node based distributed blockchain cloud architecture for IoT," *IEEE Access*, vol. 6, pp. 115–124, 2018.
- [13] C. Giannelli, P. Bellavista, D. Scotece, "Software defined networking for quality-aware management of multi-hop spontaneous networks," in *Proc. Int. Conf. Comput., Netw. Commun.*, 2018, pp. 561–566.
- [14] P. Bellavista, A. Dolci, and C. Giannelli, "MANET-oriented SDN: Motivations, challenges, and a solution prototype," in *Proc. 19th IEEE Int. Symp. World Wireless, Mobile Multimedia Netw.*, 2018, pp. 14–22.
- [15] P. Bellavista, C. Giannelli, T. Lagkas, and P. Sarigiannidis, "Quality management of surveillance multimedia streams via federated SDN controllers in Fiwi-iot integrated deployment environments," *IEEE Access*, vol. 6, pp. 21324–21341, Apr. 2018.
- [16] *Open Netw. Found., OpenFlow*. 2020. [Online]. Available: <https://www.opennetworking.org/sdn-resources/openflow>
- [17] X. Li, D. Li, J. Wan, C. Liu, and M. Imran, "Adaptive transmission optimization in SDN-based Industrial Internet of Things with edge computing," *IEEE Internet Things J.*, vol. 5, no. 3, pp. 1351–1360, Jun. 2018.
- [18] K. Kaur, S. Garg, G. S. Aujla, N. Kumar, J. J. P. C. Rodrigues, and M. Guizani, "Edge computing in the industrial Internet of Things environment: Software-defined-networks-based edge-cloud interplay," *IEEE Commun. Mag.*, vol. 56, no. 2, pp. 44–51, Feb. 2018.
- [19] Y. Duan, W. Li, X. Fu, Y. Luo, and L. Yang, "A methodology for reliability of WSN based on software defined network in adaptive industrial environment," *IEEE/CAA J. Automatica Sinica*, vol. 5, no. 1, pp. 74–82, Jan. 2018.
- [20] C. Qiu, F. R. Yu, H. Yao, C. Jiang, F. Xu, and C. Zhao, "Blockchain-based software-defined industrial Internet of Things: A dueling deep Q-learning approach," *IEEE Internet Things J.*, vol. 6, no. 3, pp. 4627–4639, Jun. 2019.
- [21] T. Grandison and M. Sloman, "A survey trust internet applications," *IEEE Commun. Surveys Tut.*, vol. 3, no. 4, pp. 2–16, Fourth Quarter 2000.
- [22] A. Visan, F. Pop, and V. Cristea, "Decentralized trust management in peer-to-peer systems," in *Proc. 10th Int. Symp. Parallel Distrib. Comput.*, Jul. 2011, pp. 232–239.
- [23] F. Pop, V. Cristea, N. Bessis, and S. Sotiriadis, "Reputation guided genetic scheduling algorithm for independent tasks in inter-clouds environments," in *Proc. 27th Int. Conf. Adv. Inf. Netw. Appl. Workshops*, Mar. 2013, pp. 772–776.
- [24] S. Buchegger and J.-Y. Le Boudec, "A robust reputation system for peer-to-peer and mobile ad-hoc networks," in *Proc. 2nd Workshop Econ. Peer-to-Peer Syst.*, Jun. 2004, pp. 1–6.
- [25] B. Zhang, Z. Huang, and Y. Xiang, "A novel multiple-level trust management framework for wireless sensor networks," *Comput. Netw.*, vol. 72, pp. 45–61, 2014.
- [26] C. Esposito, A. Castiglione, and F. Palmieri, "Information theoretic-based detection and removal of slander and/or false-praise attacks for robust trust management with Dempster-Shafer combination of linguistic fuzzy terms," *Concurr. Comput. Pract. Experience*, vol. 30, no. 3, Feb. 2018, Art. no. e4302.
- [27] J. Lin, "Divergence measures based on the Shannon entropy," *IEEE Trans. Inf. Theory*, vol. 37, no. 1, pp. 145–151, Jan. 1991.
- [28] F. Bonomi, R. Milito, J. Zhu and S. Addepalli, "Fog computing and its role in the Internet of Things," in *Proc. MCC Workshop Mobile Cloud Comput.*, Aug. 2012, pp. 13–16.
- [29] White Paper: ETSI's Mobile Edge Computing initiative explained. 2018. [Online]. Available: https://portal.etsi.org/Portals/0/TBpages/MEC/Docs/Mobile-edge_Computing_-_Introductory_Technical_White_Paper_V1%2018-09-14.pdf
- [30] K. Wolter, A. Avritzer, M. Vieira, and A. Van Moorsel, Eds., *Resilience Assessment and Evaluation of Computing Systems*. Berlin, Germany: Springer, 2002.
- [31] T. Mens *et al.*, *Studying Evolving Software Ecosystems Based on Ecological Models* (Evolving Software Systems). Berlin, Germany: Springer, 2004, pp. 297–326.
- [32] A. Bondavalli, A. Ceccarelli, P. Lollini, L. Montecchi, and M. Mori, "System-of-systems to support mobile safety critical applications: Open challenges and viable solutions," *IEEE Syst. J.*, vol. 12, no. 1, pp. 250–261, Mar. 2018.
- [33] H. Müller and N. Villegas, "Runtime evolution of highly dynamic software," in *Evolving Software Systems*. Berlin, Germany: Springer, 2014, pp. 229–264.
- [34] M. C. Huebscher and J. A. McCann, "A survey of autonomic computing - degrees, models, and applications," *ACM Comput. Surv.*, vol. 40, no. 3, pp. 1–28, Aug. 2008.
- [35] M. Salehie and L. Tahvildari, "Self-adaptive software: Landscape and research challenges," *ACM Trans. Auton. Adaptive Syst.*, vol. 4, no. 2, 2009, Art. no. 14.
- [36] D. Weyns, M. U. Iftikhar, D. G. de la Iglesia, and T. Ahmad, "A survey of formal methods in self-adaptive systems," in *Proc. 5th Int. C* Conf. Comput. Sci. Software Eng.*, 2012, pp. 67–79.
- [37] C. Murphy, G. Kaiser, I. Vo, and M. Chu, "Quality assurance of software applications using the in Vivo testing approach," in *Proc. Int. Conf. Software Testing Verification Validation*, Apr. 2009, pp. 111–120.
- [38] M. Lahami, M. Krichen, and M. Jmaiel, "Safe and efficient runtime testing framework applied in dynamic and distributed systems," *Sci. Comput. Programming*, vol. 122, pp. 1–28, 2016.
- [39] ETSI OSM Community White Paper: *Open Source MANO*. 2017. [Online]. Available: <https://osm.etsi.org/images/OSM-Whitepaper-TechContent-ReleaseTWO-FINAL.PDF>
- [40] G. Carella, M. Pauls, T. Magedanz, M. Cilloni, P. Bellavista, and L. Foschini, "Prototyping NFV-based multi-access edge computing in 5G ready networks with open baton," *IEEE Conf. Netw. Softwarization*, pp. 1–4, 2017.
- [41] White Paper: *ETSI's NFV*. 2017. [Online]. Available: https://portal.etsi.org/NFV/NFV_White_Paper_5G.pdf
- [42] P. Bellavista, C. Giannelli, and D. D. P. Montenero, "A reference model and prototype implementation for SDN-based multi layer routing in fog environments," *IEEE Trans. Netw. Service Manage.*, vol. 17, no. 3, pp. 1460–1473, Sep. 2020.
- [43] A. Cicchetti *et al.*, "CHESS: A model-driven engineering tool environment for aiding the development of complex industrial systems," in *Proc. 27th IEEE/ACM Int. Conf. Automated Software Engineering*, 2012, pp. 362–365.
- [44] L. Montecchi, P. Lollini, and A. Bondavalli, "Dependability concerns in model-driven engineering," in *Proc. 14th Int. Symp. Object/Compon./Service-Oriented Real-Time Distrib. Comput. Workshops*, Newport Beach, CA, USA, 2011, pp. 254–263.
- [45] L. Montecchi, P. Lollini, and A. Bondavalli, "A reusable toolchain for automated dependability evaluation," in *Proc. 7th Int. Conf. Performance Eval. Methodol. Tools*, Torino, Italy, 2013, pp. 298–303.
- [46] A. Avizienis, J.-C. Laprie, B. Randell, and C. Landwehr, "Basic concepts and taxonomy of dependable and secure computing," *IEEE Trans. Dependable Secure Comput.*, vol. 1, no. 1, pp. 11–33, Jan.–Mar. 2004.
- [47] A. Bondavalli, I. Mura, S. Chiaradonna, R. Filippini, S. Poli, and F. Sandrini, "DEEM: A tool for the dependability modeling and evaluation of multiple phased systems," in *Proc. Int. Conf. Dependable Syst. Netw.*, New York, NY, USA, 2000, pp. 231–236, doi: [10.1109/ICDSN.2000.857541](https://doi.org/10.1109/ICDSN.2000.857541).

- [48] É. Masson and C. Gransart, "Cyber security for railways—A huge challenge—Shift2Rail perspective," in *Proc. Int. Workshop Commun. Technologies Veh.*, 2017, pp. 97–104.
- [49] H. Li *et al.*, "Improving rail network velocity: A machine learning approach to predictive maintenance," *Transp. Res. Part C: Emerg. Technol.*, vol. 45, pp. 17–26, 2014.
- [50] R. B. Faiz and E. A. Edirisinghe, "Decision making for predictive maintenance in asset information management," *Interdisciplinary J. Inf., Knowl. Manage.*, vol. 4, no. 1, pp. 23–36, 2009.
- [51] U. Olsson and U. Espling, "Part I. A framework of partnering for infrastructure maintenance," *J. Qual. Maint. Eng.*, vol. 10, no. 4, pp. 234–247, 2004.
- [52] Particle, *The 2019 State IoT*, Report, 2019. [Online]. Available: <https://www.particle.io/solutions/2019-state-of-iot-report/>
- [53] Z. Sheng, C. Mahapatra, C. Zhu, and V. C. M. Leung, "Recent advances in industrial wireless sensor networks toward efficient management in IoT," *IEEE Access*, vol. 3, pp. 622–637, 2015.
- [54] Market Research Future, *Predictive Maintenance (PdM) Market Res. Rep. - Global Forecast till 2024*, Report MRF/ICT/1754-CR, Sep. 2019. [Online]. Available: <https://www.marketresearchfuture.com/reports/predictive-maintenance-market-2377>
- [55] E. Androulaki *et al.*, "Hyperledger fabric: A distributed operating system for permissioned blockchains," in *Proc. 13th EuroSys Conf.*, 2018, pp. 1–15.



Andrea Ceccarelli received the Ph.D. degree in informatics and automation engineering from the University of Florence, Florence, Italy, in 2012.

He is currently a Research Associate of computer science with the University of Florence. His primary research interests include design, monitoring and experimental evaluation of dependable and secure systems, and systems-of-systems. His scientific activities originated more than 100 papers which appeared in international conferences, workshops, and journals.



Marcello Cinque received the Ph.D. degree in computer and systems engineering from the University of Naples, Naples, Italy, in 2006.

He is currently an Associate Professor of computer engineering with the Department of Electrical Engineering and Information Technology, University of Naples Federico II, Naples, Italy. He is a coauthor of more than 100 papers in peer-reviewed journals and conference proceedings. His research interests include the design and assessment of dependable critical computer systems.

Dr. Cinque has been involved in the organizing committees of several international conferences on dependable computing, including the IEEE European Dependable Computing Conference and International Conference on Dependable Systems and Networks.



Christian Esposito (Member, IEEE) received the PhD degree in computer engineering and automation from the University of Napoli Federico II, Italy, in 2009.

He is currently a Tenured Assistant Professor with the Department of Computer Science with the University of Salerno, Fisciano, Italy, and received the National Qualification in Italy as an Associate Professor of computer engineering and computer science, in 2017 and 2018, respectively. His main research interests include positioning systems, reliable and

secure communications, blockchain applications, multiobjective optimization, and game theory.

Mr. Esposito regularly serves as a reviewer in journals and conferences in the field of distributed, secure, and dependable systems. He is an Associate Editor for the IEEE ACCESS and Guest Editor for many special issues in main international journals.



Luca Foschini (Senior Member, IEEE) received the Ph.D. degree in computer science engineering from the University of Bologna, Bologna, Italy, in 2007.

He is currently an Associate Professor of computer engineering with the University of Bologna. His research interests span from integrated management of distributed systems and services to mobile crowd-sourcing/-sensing, from infrastructures for the deployment of Industry 4.0 solutions to fog/edge cloud systems.

Dr. Foschini is serving as a Voting Member and awards committee chair for the IEEE Communication Society in Europe, Middle East, and Africa Board.



Carlo Giannelli (Senior Member, IEEE) received the Ph.D. degree in computer engineering from the University of Bologna, Bologna, Italy, in 2008.

He is currently an Associate Professor of computer science with the University of Ferrara, Ferrara, Italy. His primary research activities focus on industrial Internet of Things, software-defined networking, blockchain technologies, edge computing, location-based services, heterogeneous wireless technology integration, and hybrid infrastructure/ad hoc and spontaneous multipath multihop networking

environments.



Paolo Lollini received the Ph.D. degree in computer science and applications from the University of Firenze, Florence, Italy, in 2006.

He is currently an Assistant Professor with the Department of Mathematics and Computer Science, University of Firenze. He has been continuously participating in European funded projects since 2002. His research interests include the modeling and evaluation of performability and resiliency attributes of large-scale critical infrastructures, IoT, and systems of systems.