Safety of Fog-based Industrial Automation Systems

Nitin Desai
Mälardalen University
Västerås, Sweden
nitin.desai@mdh.se

Sasikumar Punnekkat
Mälardalen University
Västerås, Sweden
sasikumar.punnekkat@mdh.se

ABSTRACT

The Fog computing paradigm employing multiple technologies is expected to play a key role in a multitude of industrial applications by fulfilling futuristic requirements such as flexible and enhanced computing, storage, and networking capability closer to the field devices. While performance aspects of the Fog paradigm has been the central focus of researchers, safety aspects have not received enough attention so far. In this paper, we identify various safety challenges related to the Fog paradigm and provide specific safety design aspects as a step towards enhancing safety in industrial automation scenarios. We contextualize these ideas by invoking a distributed mobile robots use-case that can benefit from the use of the Fog paradigm.

CCS CONCEPTS

• **Computer systems organization → Embedded and cyber-physical systems**; **Fault-tolerant network topologies**; *Robotics*; Robotic autonomy.

KEYWORDS

Safety, Fog computing, Mobile robots, Industrial automation

ACM Reference Format:
Nitin Desai and Sasikumar Punnekkat. 2019. Safety of Fog-based Industrial Automation Systems. In *Workshop on Fog Computing and the IoT (IoT-Fog '19), April 15–18, 2019, Montreal, QC, Canada.* ACM, New York, NY, USA, 5 pages. https://doi.org/10.1145/3313150.3313218

1 INTRODUCTION

Cloud computing has been a key driver for many business domains over the past decade and together with latest advances in AI and machine learning it is bringing revolutionary transformations in the industry. Fog computing, a term coined by Cisco in 2012, is a distributed computing paradigm, that empowers the network devices at different hierarchical levels with various degrees of computational and storage capability [2]. It strives to *extend* the capabilities of the cloud (rather than to replace it) to cater to more stringent and extra-functional industrial automation and robotics requirements such as ultra reliable low latency communications through time sensitive networking (TSN) [15], scalable and flexible service provisioning at the desired QoS for distributed applications through virtualized control [18], and support for heterogeneous devices such as routers, gateways, and access points.

The OpenFog Reference Architecture for fog computing has been adopted as an official standard by the IEEE Standards Association (IEEE-SA). The new standard, known as IEEE 1934, relies on the reference architecture as a universal technical framework that enables the data-intensive requirements of the Internet of Things (IoT), 5G and artificial intelligence (AI) applications [1].

The Fog paradigm is an amalgamation of a host of technologies supporting multiple attributes[2]. These attributes *per se* are safety-neutral i.e., they do not in themselves impact the safety of a system positively or negatively. It is therefore the specificity of an application that dictates how these attributes interplay among each other in impacting system safety.

Whilst recent research in the Fog domain has targeted primarily performance-centric goals [2, 3, 6, 16–18], safety-centric research has not received the same treatment. Specifically, we need research focusing on software architectures of Fog systems that run safety-critical applications such as robotics and industrial automation. Although in a sense, the Fog paradigm can be seen as an attempt to enhance certain safety-related aspects of the cloud (e.g., reducing latencies to ensure safety-critical tasks are completed on time), there are challenges unique to Fog systems.

Research on Fog computing is still in its nascency and works pertaining specifically to safety of Fog architectures are scarce. In [14] the authors discuss a fault tolerant framework for autonomous systems through safety monitors by the generation of safety rules based on safety margins. In [11], the authors propose skill-based architecture for mobile robots, together with a novel risk assessment and decision-making model.

In this paper, for illustration purposes we use a typical industrial automation setting that employs mobile robots to accomplish various missions. These Mobile robots themselves have limited computation and communication capabilities, and usage of cloud is being explored by the industry. However, moving information to cloud needs larger communication bandwidth and often jeopardizes the predictability guarantees essential for these class of applications. The introduction of the Fog paradigm is expected to contribute to the reduction of latencies, greater flexibility in task allocations and scalable deployments, and presents as an attractive alternative if one can resolve the safety and security concerns.

The rest of the paper is structured as follows. Section 2 introduces safety considerations of a Fog-based architecture and identifies challenges. In section 4 we provide the preliminary ideas towards development of a framework to tackle such challenges by proposing safety design aspects. Section 5 provides a concrete use-case to

IoT-Fog '19, April 15–18, 2019, Montreal, QC, Canada

ACM ISBN 978-1-4503-6698-4/19/04…$15.00
https://doi.org/10.1145/3313150.3313218

Table 1: Fog attributes and potential threats to safety

Fog attribute	Potential threat to safety
Virtualized hardware control	High control loop latency
Real-Time (RT) response	RT tasks could be serviced non optimally in Fog-cloud
Scalable deployments	System unable to support the scale of deployments
Data filtering and aggregation	Safety-relevant data being missed
Seamless resource management	Unavailability of resources for safety-critical services
Mobility management	Incorrect virtual clusters
Wireless communication	Non-determinism in execution of safety function
Security and data privacy	Malicious Fog nodes posing safety risks
Time synchronization	Safety-critical deadline misses

put these ideas into perspective. Conclusions and future research directions are provided in section 6.

2 SAFETY CONSIDERATIONS FOR THE FOG

Safety certification of a system is typically done at design time prior to deployment for a specified system configuration under specified operating conditions. This is best suited for static systems whose configurations remain unchanged during the course of operation. Fog based systems by their very definition are *dynamic*, *adaptive*, *re-configurable*, and need to *scale* as required by the applications that benefit from the Fog paradigm.

In the present mobile robotic context, safety is a function of uncertainty in both the robot's dynamics and those of its surroundings. The critical task is to ensure a mobile robot's safety when operating in close proximity with a rapidly *evolving* and *stochastic* environment [12]. In such a dynamic context, the probability of safety hazards is higher than for a purely static system (which can be safety certified during design stage).

Consequently, confidence in the safety of autonomous systems, e.g., assistive robots, medical robots, or co-workers, is the main barrier to their deployment in everyday life. In Table 1 we present some of the Fog attributes [2][3][15] and their potential threats to safety. The attributes *per se* do not cause any safety risk but certain safety-critical applications running in the Fog could require *guaranteed* bounds on these attributes. Unless such guarantees can be ensured, there always exists a risk. For instance, time synchronization between robots could be a safety-critical requirement. Safety cannot be guaranteed unless time synchronization is also guaranteed.

3 CHALLENGES TO SAFETY ASSURANCE

Here we present some of the key challenges relevant to safety for the Fog paradigm.

3.1 C1 - Domain-specific safety context

Safety-centric design of Fog systems is a complex and challenging pursuit since it is not a static phenomenon. Consequently, to assume the same safety context (e.g., motion safety) for two different domains can prove erroneous. For instance, an emergency-stop function for an automotive use-case cannot be applied directly to an aerospace use-case without dire consequences.

3.2 C2 - Evolving safety goals

Though Fog presents a great opportunity w.r.t. reconfigurability and adaptability, the designers should be careful not to overlook the associated safety risks. A current research challenge is to ensure safety as a consequence of autonomous decision making in robotic control. This is relevant to collaborative robotic systems in general and Fog-based robotics in particular. Adapting to new, evolving safety requirements on the fly at run time is needed.

3.3 C3 - Safety relevant data extraction

A general characteristic of Fog systems is the large amount of data generated at the end points (e.g., robots generating images, sensor data). Efficient data filtering and aggregation mechanisms are necessary to ensure that safety-critical functions are evaluated accurately. At the same time one should make sure that the safety relevant data is not being filtered or aggregated upon.

3.4 C4 - Resource availability guarantees

A primary performance goal is to ensure seamless resource provisioning across the Fog-cloud continuum. For safety critical applications, it is imperative to have the right resources, at the right time. Hence, timing *Guarantees* need to be put in place at the networking level to ensure services are available at all times for the critical tasks.

3.5 C5 - Security

Safety and security lie at an intersection and hence we need identification of potential security threats such as malicious attacks from rogue Fog nodes that can disrupt safety-critical requirements such as clock synchronization quality. Hence, appropriate mitigation techniques are needed.

3.6 C6 - Lack of relevant safety standards

Two specific robotic safety standards are ISO 10218:2011 for robots in industrial environments [8] and ISO/TS 15066:2016 [10] for collaborative robots.

However, very few robots have been safety certified. For instance, the technical documentation of the UR5 from Universal Robots specifies that 15 safety functions have been tested by the TÜV (*Technischer Überwachungs-Verein*) in accordance with the EN ISO 13849:2008 PL d [9], and EN ISO 10218-1:2011, Clause 5.4.3 [8]. It is important to note that this certificate only validates the presence of a safety function (clause 5.4.3), with PL d (equivalent to the medium level SIL 2 in [7]). This does not guarantee safety in the context of a given task and environment [5].

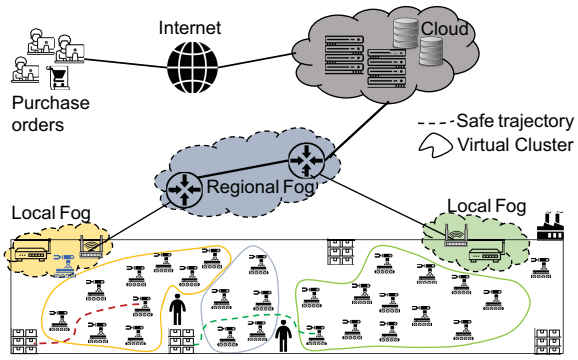

Figure 1: A typical Fog-based factory automation setting.

Therefore, safety certification for collaborative, dynamic autonomous systems based on Fog still remains as an important challenge.

Fig. 1 depicts a high-level view of a Fog-based distributed mobile robot scenario. Each Fog node has its virtual cluster of robots as shown. These clusters represent the set of robots that are designated to the Fog node. These can however change dynamically based on service requirements. The robots (which can turn into Fog nodes) have on-board sensors and cameras which report visual and sensory data about the immediate surroundings of the robot. The safe trajectory is a collision free path that has been assigned to the robots by the Fog node.

4 FOG SAFETY DESIGN ASPECTS

The principal research question we are pursuing to counter the challenges presented in section 3 is as follows:

"How can the Fog system architecture ensure that safety-critical functions execute as expected in the presence of diverse and uncertain operating conditions?"

The word *diverse* can have multiple connotations of which two particularly relevant ones from a safety viewpoint are:

(1) A diverse set of *software services* each robot must handle (e.g., pick and place functions for heavy and fragile goods requiring different system configurations in control, motion etc.).
(2) Adapting to diverse (and sometimes unexpected) *operating conditions* such as sensor uncertainties or failures.

A fundamental goal is to establish a conceptual safety framework motivated by the challenges addressed previously. The purpose of a safety framework is to abstract Fog safety requirements and enable multiple applications (and their safety contexts) to be represented. The idea of having *safety-as-a-service* can provide opportunities to explore safety primitives in the Fog more concretely.

The elements in the design of the said framework must include attributes specific to the Fog paradigm and must tackle these challenges.

We now delineate some of the key safety design aspects that will contribute to the formulation of a conceptual safety framework for Fog software architectures.

The ordering of the safety design aspects are in no way representative of their relevance or importance.

4.1 S1 - Safety state and context identification

As shown in Fig. 2, the safety state monitor provides the decision module with relevant data to decide if operational condition is one of Normal, Warning, or Error states. Each application instance running in the Fog-cloud should have a safety context attribute as the safety function execution heavily depends on the type of application. E.g., the force threshold for the end effectors of a surgical robot is much smaller than that of a robotic arm in an automotive assembly unit. Challenges C1 and C3 are addressed here.

4.2 S2 - Safety criticality levels

The safety-critical functions running in the Fog must be accorded multiple priority levels. Having a static priority level will not fulfill diverse service needs satisfactorily. This aspect addresses challenge C2.

4.3 S3 - Safety-critical resource availability

In the resource allocation for applications, it is necessary to consider the case when a resource is unavailable for safety-critical functions. Such functions must have guaranteed resources available at all times even when the system is swamped with other non safety-critical computation tasks. This safety aspect addresses challenge C4. The time triggered paradigm which provides a deterministic communication backbone (TTEthernet / TSN) can provide time guarantees to ensure resource availability [15].

4.4 S4 - Fault detection and recovery

A characteristic of the Fog node [15] is to have statistical data on end devices. However, if the data provided has uncertainties or is faulty, it paves the way for erroneous decisions and can jeopardize safety. E.g., sensor malfunction can lead to false positive errors. Localization of such errors is an essential safety primitive that should be addressed in the Fog. In addition, sensor wear and tear must be periodically sent to the Fog. However, this is more generic and doesn't target a specific challenge.

4.5 S5 - Data mining

Automation applications with sensors as end devices generate a lot of data. It is imprudent to process such vast quantities of data all the time. Past safety hazards should be considered in the evaluation of risk to safety from the system states. Proper categorization as low, medium and high risk is needed. If no hazards are evident for a specific operating condition and for a certain duration of time, less data can be used. The other benefit from this design primitive is the reduced use of data storage. This tackles challenge C3.

4.6 S6 - Security mechanisms

Malicious nodes within the Fog-cloud continuum can pose a safety and security threat [4]. Robust security mechanisms such as industry standard encryption techniques need to be considered for authentication and identification of all Fog entities. One solution is secure boot [13] that prevents over-the-air firmware updates

from unrecognized sources (e.g., malicious nodes). This helps tackle challenge C5.

4.7 S7 - Active mitigation

In the event of a safety hazard, the safety application must decide the best course of action to mitigate the impact of the hazard without causing further damage of system performance and operation. Active intervention mechanisms to provide fail-safe procedures play a pivotal role. E.g., Applying brakes when the robot has breached the safe stopping distance. This tackles challenge C4.

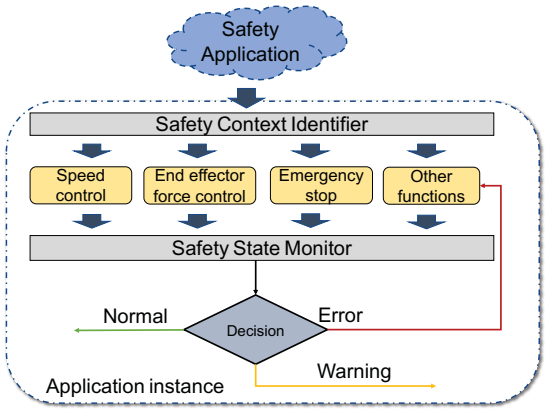

Figure 2: A safety application instance executing in the Fog.

Fig. 2 shows a safety application running in the Fog. The application oversees the general safety requirements of the system by identifying the relevant safety contexts for specific safety-critical functions (such as obstacle/collision avoidance). The monitor keeps track of such functions in order to decide the safety state of the system - normal, warning, or error - based on rules that are updated dynamically. In the event of an error state, the corresponding safety functions are identified and appropriate actions taken.

5 ILLUSTRATIVE USE-CASE

We are currently developing a distributed mobile robot use-case, which is industry relevant within the smart manufacturing domain. We intend to use this use-case to elaborate and concretize the safety design aspects described in the paper.

Below are key components of this use-case.

Mobile Robot. The robots perform various pick and place tasks using the robot arm as well as move freely within the factory floor. They are equipped with on-board *sensors* (LIDAR, radar, cameras). The physical attributes of an individual mobile robot are characterized by the CPU processing, RAM, storage, and energy. These robots are connected to *Fog nodes* or more generally, the Fog through the IEEE 802.1 TSN standard. The Fog designates *tasks* to the robots within its *virtual cluster* (see Fig. 1) and monitors completion of robotic missions. In addition, at any given instant, the Fog node has knowledge of the resources that are provided to the robots [15] as well as the safety states.

Fog-cloud continuum. Deterministic connectivity between the robots (IoT layer) and the Fog is provided by the TSN standard [15]. Safety applications run on the Fog-cloud to ensure the system remains within safe states. At any instant of time, the Fog node can request safety state information from any device across within the continuum.

Mode of operation. The motion control algorithms in the application make use of sensor data from the robots. Lidar can used to provide an accurate 3D map of the topology surrounding the robot (as in the case of automotive). However, the prohibitively high cost of a lidar sensor can be a deterrent to its use for a large scale robot deployment. Therefore, we consider high resolution cameras which capture images within the robot's field of view. One of the ideas to ensure safety is to *fuse* multiple robot fields of view to generate an integrated image in the Fog. Such an image would provide the Fog nodes with sufficient information to make real-time motion control decisions.

A concrete research problem would be to see how such a distributed processing can be performed in the Fog nodes and ultimately, its impact on motion safety in quantitative terms (such as number of collisions).

6 CONCLUSION AND FUTURE WORK

This paper presented a set of safety challenges and concrete safety design aspects towards the creation of a comprehensive safety framework that is specifically tailored to cater to the Fog computing paradigm. Our ongoing research includes the following three areas.

6.1 Fog modeling and evaluation

Our ultimate aim is the formulation of a comprehensive conceptual framework to ensure safety in the Fog software architecture. The said framework would provide the means to express safety aspects concretely in terms of a theoretical model [17, 18]. The model would be to provide a quantitative evaluation of the safety states of the system through the probability of occurrence of safety hazards considering all relevant run-time operational aspects.

6.2 Simulation support

We intend to evaluate the scenario involving unavailability of resources for functions in the application running in the Fog-cloud using iFogSim [6], which is a popular simulation tool.

Unlike resource scheduling, safety cannot be guaranteed by the notion of a optimum i.e., an optimum set of resources allocated with performance targets in mind need not be the *safest* in terms of execution of safety-critical tasks. Hence, to guarantee that the Fog architecture is designed to ensure safety across various operational conditions, existing simulators need to incorporate safety primitives as well.

ACKNOWLEDGEMENT

The research leading to these results has received funding from the European Union's Horizon 2020 research and innovation programme under the Marie Skłodowska-Curie grant agreement No. 764785, FORA—Fog Computing for Robotics and Industrial Automation.

REFERENCES

[1] 2018. IEEE Standard for Adoption of OpenFog Reference Architecture for Fog Computing. *IEEE Std 1934-2018* (Aug 2018), 1–176. https://doi.org/10.1109/IEEESTD.2018.8423800

[2] F. Bonomi, R. Milito, J. Zhu, and S. Addepalli. 2012. Fog Computing and Its Role in the Internet of Things. In *Proceedings of the First Edition of the MCC Workshop on Mobile Cloud Computing (MCC '12)*. ACM, New York, NY, USA, 13–16. https://doi.org/10.1145/2342509.2342513

[3] A.V. Dastjerdi, H. Gupta, R.N. Calheiros, S.K. Ghosh, and R. Buyya. 2016. Chapter 4 - Fog Computing: principles, architectures, and applications. In *Internet of Things*, Rajkumar Buyya and Amir Vahid Dastjerdi (Eds.). Morgan Kaufmann, 61 – 75. https://doi.org/10.1016/B978-0-12-805395-9.00004-6

[4] Y. Guan, J. Shao, G. Wei, and M. Xie. 2018. Data Security and Privacy in Fog Computing. *IEEE Network* 32, 5 (Sep. 2018), 106–111. https://doi.org/10.1109/MNET.2018.1700250

[5] J. Guiochet, M. Machin, and H. Waeselynck. 2017. Safety-critical advanced robots: A survey. *Robotics and Autonomous Systems* 94 (2017), 43 – 52. https://doi.org/10.1016/j.robot.2017.04.004

[6] H. Gupta, A. V. Dastjerdi, S. K. Ghosh, and R. Buyya. 2017. iFogSim: A Toolkit for Modeling and Simulation of Resource Management Techniques in Internet of Things, Edge and Fog Computing Environments. *Softw., Pract. Exper.* 47 (2017), 1275–1296.

[7] IEC61508. 2010. Functional safety of electrical/electronic/programmable electronic safety-related systems, IEC Std. 61508.

[8] ISO10218-1. 2011. Robots and robotic devices - safety requirements for industrial robots - part 1: Robots, International Organization for Standardization.

[9] ISO13849-1. 2006. Safety of machinery - safety-related parts of control systems - part 1: General principles for design, International Organization for Standardization.

[10] ISOTS15066. 2010. Robots and robotic devices - safety requirements for industrial robots - collaborative operation, International Organization for Standardization.

[11] A. F. Leite, A. M. Pinto, and A. Matos. 2018. A Safety Monitoring Model for a Faulty Mobile Robot. *Robotics* 7 (2018), 32.

[12] K. Leung, E. Schmerling, M. Chen, J. Talbot, J. C. Gerdes, and M. Pavone. 2018. On Infusing Reachability-Based Safety Assurance within Probabilistic Planning Frameworks for Human-Robot Vehicle Interactions. arXiv:arXiv:1812.11315

[13] Y. Liu, J. Briones, R. Zhou, and N. Magotra. 2017. Study of secure boot with a FPGA-based IoT device. In *2017 IEEE 60th International Midwest Symposium on Circuits and Systems (MWSCAS)*. 1053–1056. https://doi.org/10.1109/MWSCAS.2017.8053108

[14] M. Machin, J. Guiochet, H. Waeselynck, J. Blanquart, M. Roy, and L. Masson. 2018. SMOF: A Safety Monitoring Framework for Autonomous Systems. *IEEE Transactions on Systems, Man, and Cybernetics: Systems* 48, 5 (May 2018), 702–715. https://doi.org/10.1109/TSMC.2016.2633291

[15] P. Pop, M. L. Raagaard, M. Gutierrez, and W. Steiner. 2018. Enabling Fog Computing for Industrial Automation Through Time-Sensitive Networking (TSN). *IEEE Communications Standards Magazine* 2, 2 (June 2018), 55–61. https://doi.org/10.1109/MCOMSTD.2018.1700057

[16] S. Sarkar, S. Chatterjee, and S. Misra. 2018. Assessment of the Suitability of Fog Computing in the Context of Internet of Things. *IEEE Transactions on Cloud Computing* 6, 1 (Jan 2018), 46–59. https://doi.org/10.1109/TCC.2015.2485206

[17] S. Sarkar and S. Misra. 2016. Theoretical modelling of Fog computing: a green computing paradigm to support IoT applications. *IET Networks* 5, 2 (2016), 23–29. https://doi.org/10.1049/iet-net.2015.0034

[18] O. Skarlat, M. Nardelli, S. Schulte, M. Borkowski, and P. Leitner. 2017. Optimized IoT Service Placement in the Fog. *Serv. Oriented Comput. Appl.* 11, 4 (Dec. 2017), 427–443. https://doi.org/10.1007/s11761-017-0219-8