Complex distributed systems such as the ones induced by Internet of Things (IoT) deployments, are expected to operate in compliance to their requirements. This can be checked by inspecting events flowing throughout the system, typically originating from end-devices and reflecting arbitrary actions, changes in state or sensing. Such events typically reflect the behavior of the overall IoT system — they may indicate executions which satisfy or violate its requirements.

Supporting such validation in practice however, is challenging. Firstly, the sheer number of devices and their heterogeneity inherent in IoT systems require dedicated software architectures — typical cloud-based deployments are often not applicable. Secondly, communication particularities of the IoT domain need to be taken into account, since devices may be connected through a plethora of networking technologies exhibiting different characteristics, such as low-power wide-area networks (LPWAN). Finally, the volume and velocity of the events generated in realistic IoT systems can saturate network links and centralized processing schemes.

We present [1] a service-based software architecture and technical framework supporting runtime verification for decentralized edge-intensive systems. At the lowest level, systems we consider are comprised of resource-constrained devices connected over wide area networks generating events. Monitors are deployed on edge components, receiving events originating from end-devices and other edge nodes. Properties expressing desired system requirements are evaluated on each edge monitor in a runtime fashion. The system exhibits decentralization, since property evaluation occurs locally on an edge node, while evaluation verdicts possibly affecting satisfaction of others are propagated accordingly. We assume that requirements to be satisfied by the system under design are specified in terms of assertions in a temporal logic. We subsequently leverage results on runtime verification as its technical domain.

Our framework [1] utilizes Metric First-Order Temporal Logic (MFOTL), a formalism operating on traces of events. Our motivations for choosing the formalism are (i) its expressiveness, as MFOTL has seen applications on systems ranging from financial to cyber-physical, and (ii) the fact that it is well-defined and has well-studied theoretical semantics and properties, both rendering practical adoption and tool support easier. We contextify the framework advocated within the wide domain of IoT monitoring, but stress particularly runtime verification as its technical domain.

To provide concrete evidence of the applicability of the proposed architecture and technical framework, we first illustrate how runtime verification can be achieved in practice on a case study of a spatially-distributed parking system in a large metropolitan area, drawing from publicly available parking data from the city of Los Angeles. We then investigate the feasibility of our design to operate in resource-constrained edge computing environments over a testbed instantiation. Thereupon, we evaluate performance and capacity limits of different hardware classes, from small single-board computers to server-class data center hosts, under monitoring workloads of varying intensity, for end-devices communicating over LoRaWAN, a popular LPWAN technology.

REFERENCES