

Title: sMart EDge fabric for IoT Applications (MEDInA)

Speaker: Dr. Nabil Abdennadher, Full Professor, University of Applied Sciences Western Switzerland

This talk presents a generic platform for intelligent IoT applications based on a shareable backbone infrastructure composed of three layers: IoT objects, edge devices and cloud infrastructure. Our framework:

- delivers machine learning models (MLM) learned in the cloud over data streams collected by the edge, from IoT devices.
- supports lightweight learning algorithms that can execute on the edge and self-adapt without any synchronisation with the cloud

The platform delivers the following functionalities:

1. Coordinate application deployment from the cloud to the edge. The platform will target cloud, edge and IoT devices. IoT applications can be deployed, configured, operated and maintained, using a shared infrastructure, where several applications can coexist.
2. Continuously integrate, deploy and maintain MLMs on edge devices. Learning, which requires considerably more resources, will take place in the cloud and the learned models will be deployed at the edge. This will considerably decrease the response time and the necessary bandwidth between the cloud and edge layer, since real-time data processing will take place close to the IoT devices. With this functionality, the edge triggers the learning process, although it is performed in the cloud.
3. Use lightweight, yet powerful, machine learning models that can be setup by, and deployed on, resource constrained devices that are typically used by edge devices. This alternative is possible for some IoT applications for which the learning process is light enough to run on the edge. With this functionality, the edge is more autonomous, its intelligence is “improved” locally.

A Shareable edge/cloud platform for IoT applications

The platform targets IoT applications deployed on edge/cloud platforms. Modules are distributed between edge devices and cloud infrastructures (Figure 1). IoT applications are represented by a set of edge and cloud modules (EM: Edge Module, CM: Cloud Module) deploy on appropriate edge devices and cloud infrastructure.

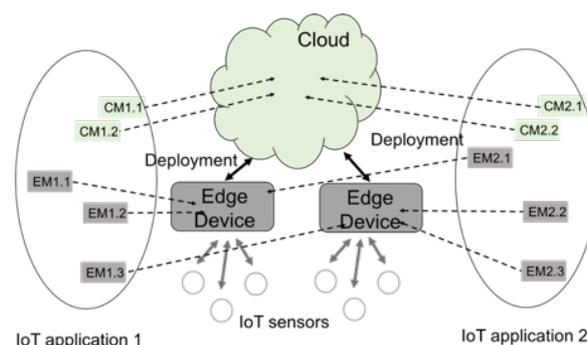


Figure 1: IoT applications deployed on a shareable hybrid edge/cloud platforms.

An edge-controlled intelligence

The main objective is to bring intelligence to the edge, using for example machine learning, such that edge devices can self-configure and react autonomously to local events. The deployment of MLMs on the edge devices will roughly follow a number of standard steps: Figure 2 gives a schematic view of the continuous integration cycle of MLMs. The IoT application generates an initial MLM from a set of training data (Figure 2, Step 1). The MLM is then deployed through the platform (Figure 2, Step 2). The performance of the deployed MLM will be continuously monitored: each time the MLM is unable to take

a reliable decision, it sends the data it failed to process to the cloud. The cloud then collects these datasets (Figure 2, Step 3) from different edge devices and triggers a new learning cycle, resulting in a new MLM which will then be pushed back to the edge devices (Figure 2, Step 4). This feature will offer a first version of the InaaS architecture where edge intelligence is controlled by the cloud.

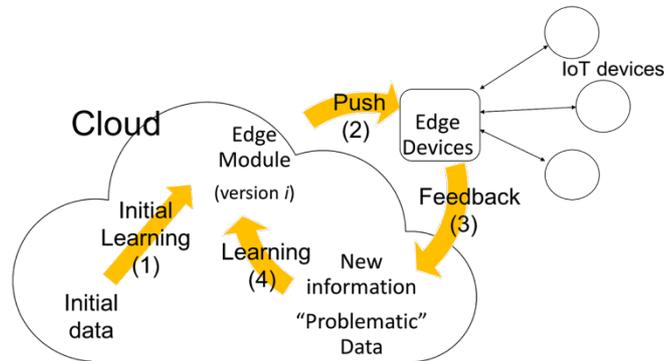


Figure 2: continuous integration architecture for MLMs.

An autonomous edge intelligence

In several cases, MLMs are generated by lightweight machine learning algorithms that do not need intensive computing, and hence can self-adapt locally by the edge devices (on-site learning, Step 5, Figure 3). This functionality can be applied to several IoT applications. We then envisage a learning scenario in which the MLMs are improved locally (on-site) as new data is generated by the devices. This feature will offer an extended version of InaaS where the edge can improve its intelligence locally.

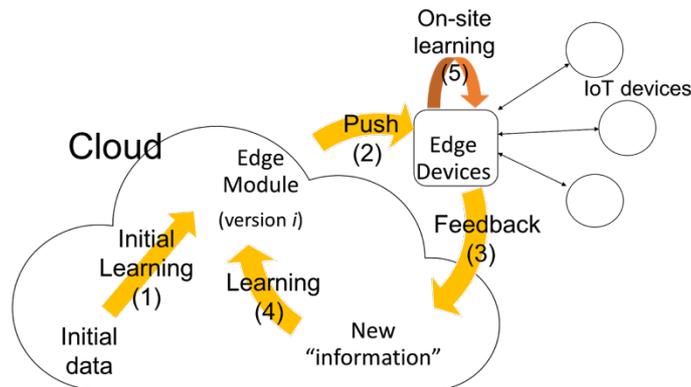


Figure 3: Continuous integration architecture for MLMs where the learning system is deployed on the cloud (step 4) and/or on the edge (step 5).