Demo: Empowering Cyber-Physical Systems with FADEX

Vittorio Cozzolino†, Aaron Yi Ding†, Ardalan Amiri Sani‡, Richard Mortier§, Dirk Kutcher⋆, Jörg Ott†

†Technical University of Munich ‡UC Irvine §University of Cambridge ⋆Huawei
{cozzolin, ding, ott}@in.tum.de, ardalan@uci.edu, richard.mortier@cl.cam.ac.uk, dirk.kutscher@huawei.com

The proliferation of smart devices in close proximity to end users has massively increased availability of data about our surroundings and hence stimulated a plethora of new services. However, it has also increased the chances of leaking sensitive and private information about end users (e.g., geolocation data, biometric signatures). Loss of trust towards a Cloud provider can lead to a user boycott and requests for deletion of the their remotely stored personal information. While many Cloud services can handle this relatively easily, it is far more cumbersome for many smart services. In fact, the current market of smart services is composed of black-box systems dependent on tight coupling between deployed hardware and the Cloud hosted software stack leaving virtually no freedom to change service provider without considerable redeployment costs.

In this demo, we present FADEX, a Xen based, edge offloading platform designed and developed to run on off-the-shelf devices. As shown in Figure 1, unikernels (we use MirageOS) are a key component allowing us to offload only the required code without additional overhead due to layering of runtimes, shared libraries or other dependencies. Therefore, we call this approach *fine-grained offloading*: each unikernel is a single-purpose appliance containing only the code required by the application, resulting in reduced image size, improved security and greater manageability. Our system is designed with two specific goals in mind: (1) offer a virtualized ecosystem on top of the edge computing infrastructure to host arbitrary services; (2) provide enhanced performance by exploiting data locality.

FADEX enables several cyber-physical systems use cases: from image processing, to data fusing, to direct access to external hardware devices. Specifically, FADEX's memory address management stack enables users to interact with sensors and actuators directly from inside virtual machines through a completely virtualized *driver stack*. The logic to control an external hardware device is contained inside the virtual machine and is independent from the host system. We thus also allow fine-grained access to hardware resources by exposing physical registers based on contract policies.

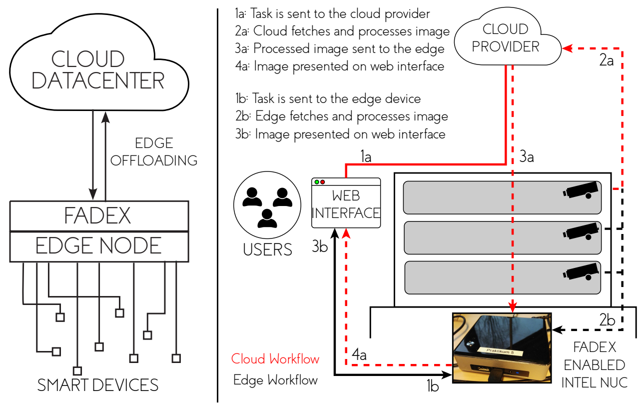

Figure 1: FADEX Architecture and Use Case

The use case selected for this demo is image processing. We consider a smart building equipped with multiple cameras on different floors. An edge node hosting FADEX is connected to multiple smart cameras and is ready to serve requests to process images or video feeds on-demand. Each user is free to decide which operation to run on the data. We replicated the depicted scenario by deploying an Intel NUC in a smart home with our system installed on it. The network connection had an uplink with an effective speed of 5 Mbps. In this test, we used the image processing unikernel configured in monochrome mode with different image sizes. Our results show that processing data locally is much faster than uploading data to the Cloud for processing. High transfer time is the main cause behind the poor Cloud performance. This is especially true when the amount of data to be processed is large.

Demo: We will setup an Intel NUC connected to a USB camera, which will be used to take pictures in real time. The collected pictures will be processed by FADEX through a selectable operations such as monochrome, resizing or normalizing. The demo attendants can interact with our system through a webpage in order to submit tasks and decide if the operation should be carried out at the edge or in the Cloud. Performance metrics will then be presented on a screen.