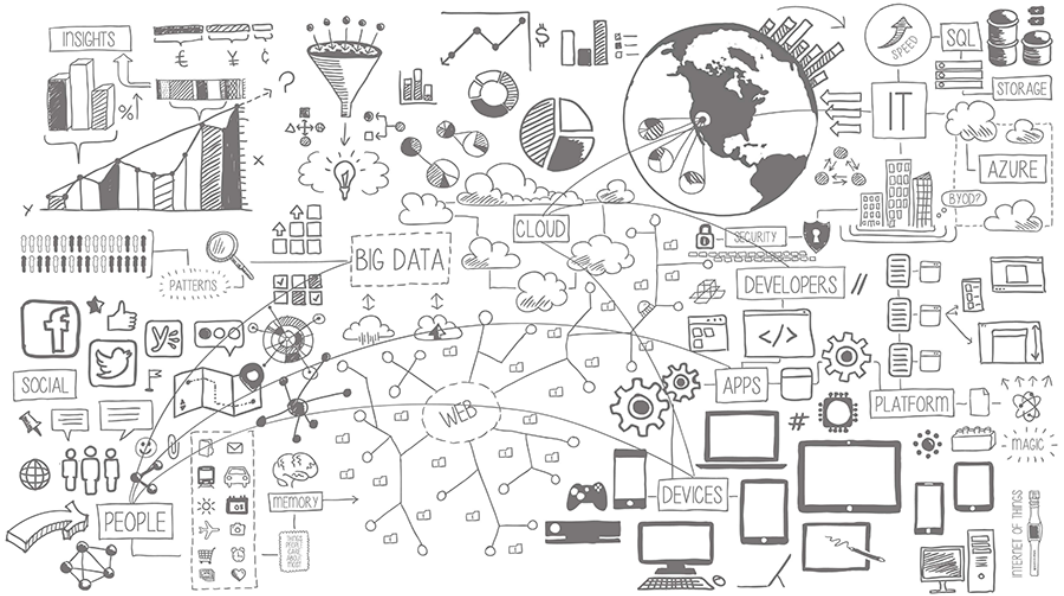
Review Open Call 2 – Stage 2 experiment COMFORT-APP

Kostas Trichias

WINGS ICT Solutions

FEC4

Bruges, Belgium, 08-10 October 2018



*Picture credit: HeroX Smart Bearing Challenge

COMFORT-APP

COMPUTATIONAL OFFLOADING FOR IOT ENABLED APPLICATIONS

Background & Motivation

WINGS Backround

- Strong backroud in Edge Mobile/Cloud & Internet of Things platforms
- Multiple applications, platforms and demos build with remote sensors
- WINGS “smart city” and “water management” solutions rely on IoT devices feeding data to cloud servers

WINGS Motivation

- Develop an algorithm to dynamically optimize resource utilization and increase IoT devices energy efficiency to improve the WINGS platforms
- Provide guaranteed QoS metrics in varying conditions, to be able to utilize IoT and sensor devices to an extended context with stringent reuirements (e.g. 5G apps)

Experiment Description

CONCEPT & OBJECTIVES

- *Computational Offloading* increasingly important in IoT paradigm.
- COMFORT-APP aims to:
  - Alleviate computational limitations at the network edge.
  - Minimize mobile devices' energy consumption.
  - Guarantee a certain level of QoS for the end users.
  - Load Balancing between MEC and Cloud.

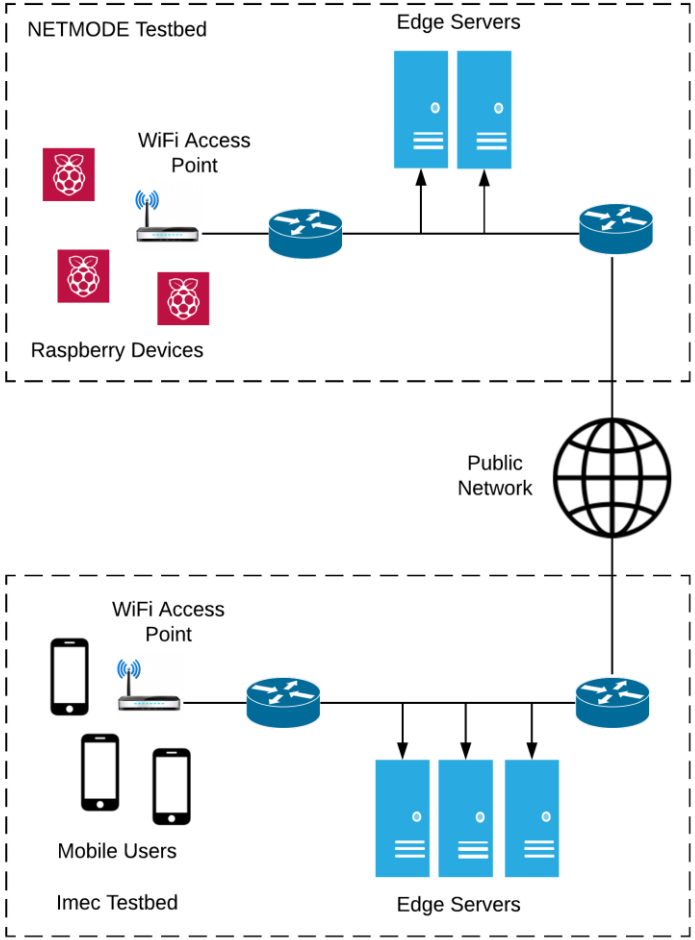
Experiment Description

SET-UP

- A simple computational offloading framework.
- Target Applications: Tesseract OCR Engine and Google's Tensorflow –based.

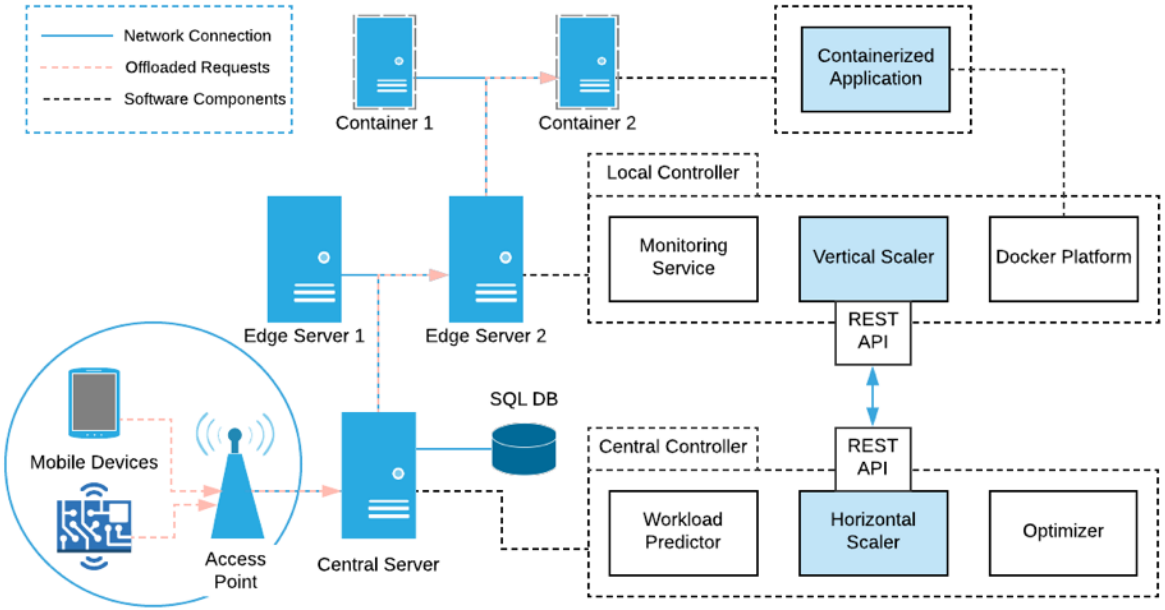
✓ NETMODE: Raspberry Pis with cameras and powerbanks.

✓ w-iLab.t : DSS mobile nodes & snapshots.



Experiment Description

MEC ARCHITECTURE



Central Controller (CC)

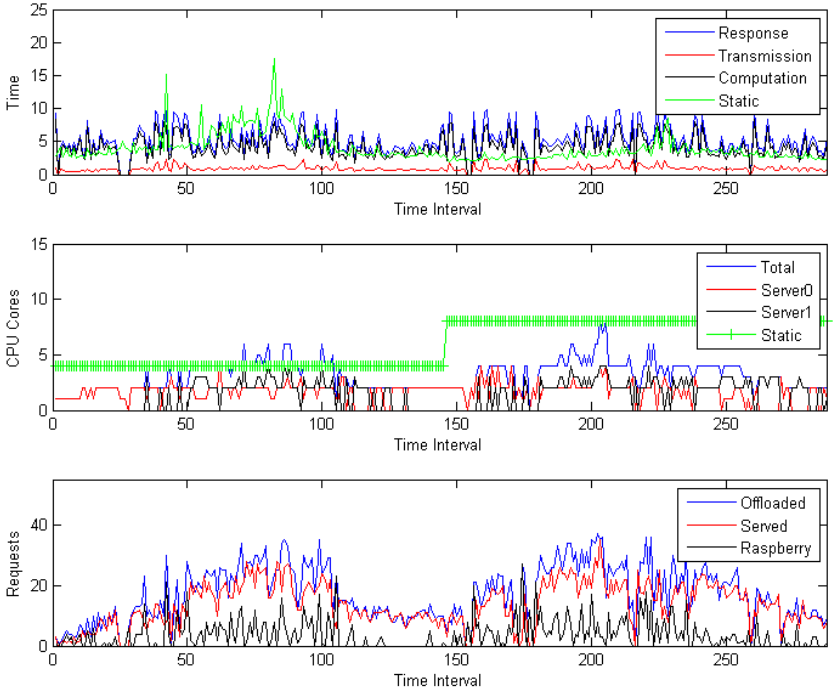
1. collects data
2. predicts workload
3. Horizontal scaling

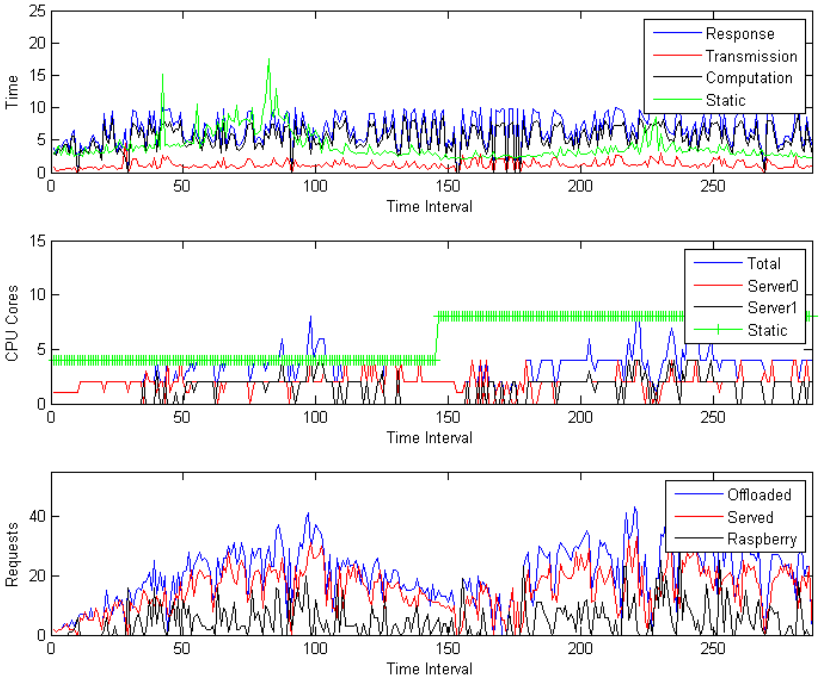
Local Controller (LC)

1. realizes CC's decisions
2. Vertical Scaling
3. controls CAs (Docker)

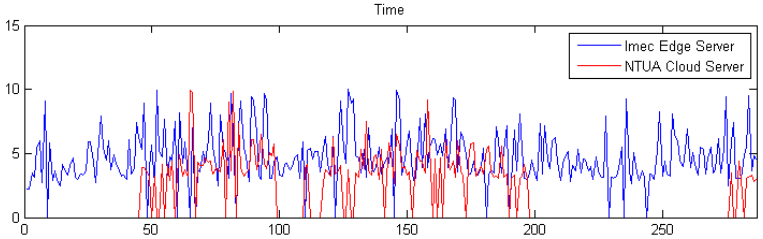
Containerized Applications (CAs)

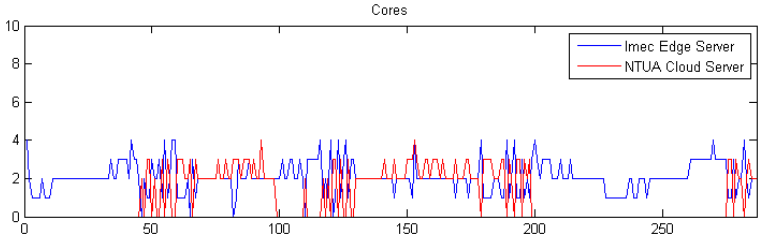
MEC Example

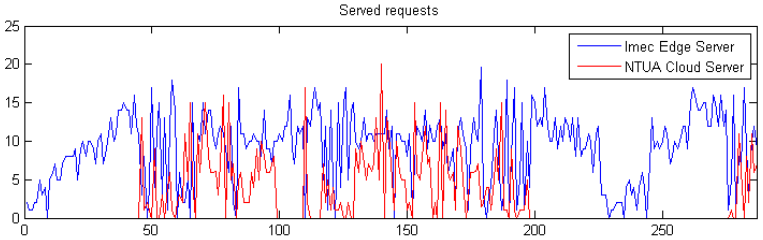

Time
Response
Transmission
Computation
Static
Time Interval
CPU Cores
Total
Server0
Server1
Static
Time Interval
Requests
Offloaded
Served
Raspberry
Time Interval

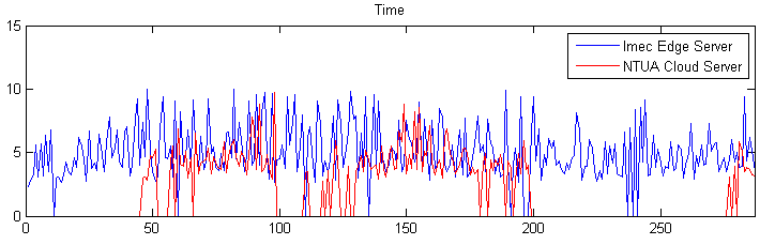

Time
Response
Transmission
Computation
Static
Time Interval
CPU Cores
Total
Server0
Server1
Static
Time Interval
Requests
Offloaded
Served
Raspberry
Time Interval

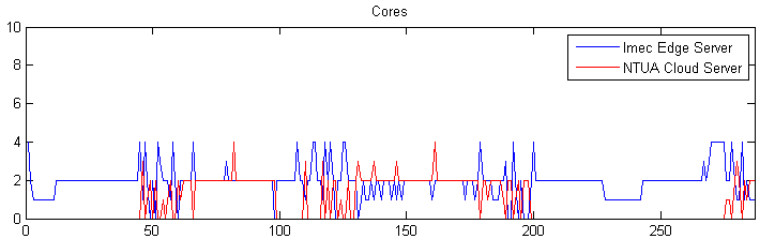
Load Balancing

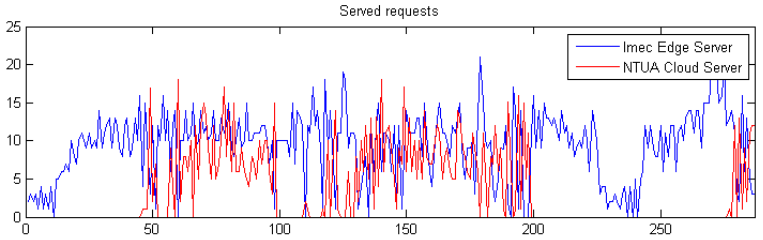






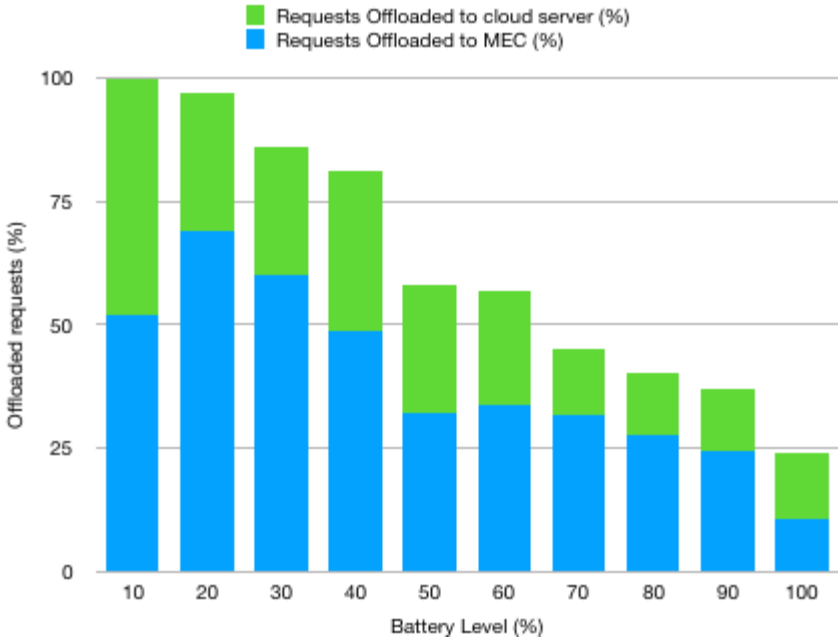




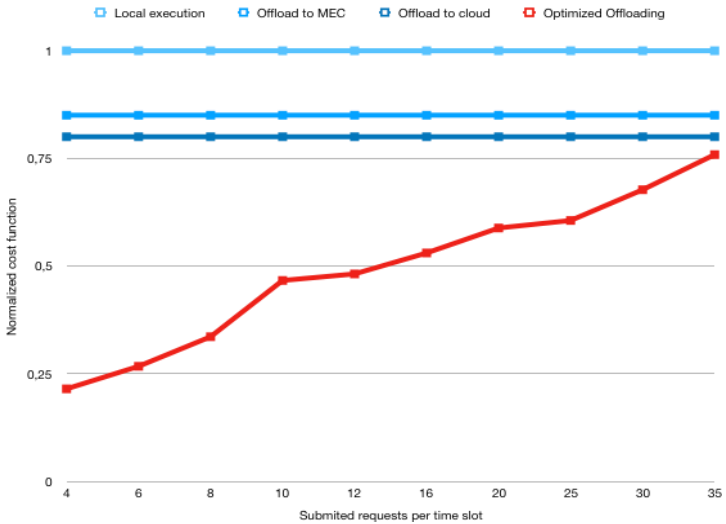


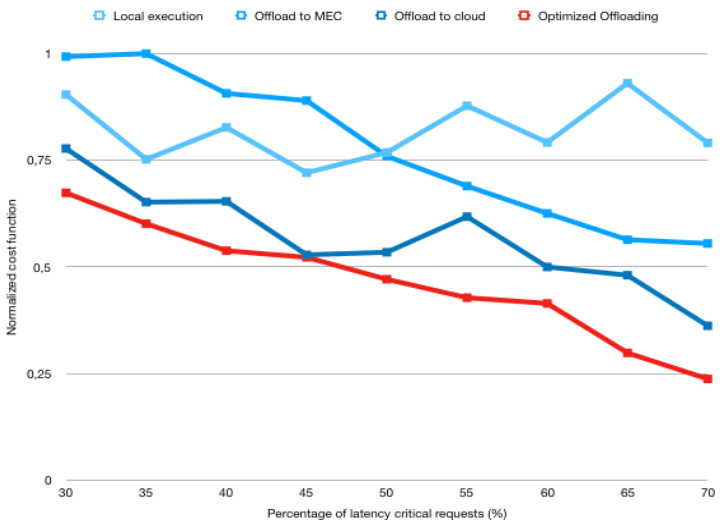
Dynamic offloading

- Task offloading based on remaining end-device energy, takes optimal offloading decision and protects the device
- Larger portion of tasks can be executed locally for high battery levels
- Significant energy savings for energy-constrained devices
- Task allocation between MEC and cloud varies based on their load, with a preference to the MEC



Dynamic offloading





- COMFORT-APP dynamic offloading clearly outperforms all static offloading mechanisms
- Much lower cost (resources, computational, energy) in order to provide the same QoS, especially for a low number of requests
- Dynamic offloading is also much better for time-critical applications, meeting the required QoS with decreased cost for the system.

Lessons Learned

- Horizontal and Vertical Scaling of Edge Servers is essential for **guaranteeing the QoS metrics of time or mission critical applications**.
- Up to **50% reduction of the energy consumption** of end devices can be achieved.
- Horizontal Scaling enables **Load Balancing between MEC and Cloud**.
- Dynamic resource allocation **prevents over / under- provisioning of edge servers**.
- Dynamic MEC offloading significantly outperforms static offloading in terms of **averages task execution latency and optimized use of resources**.
- The retention of the desired **QoS level is attainable with a minimized cost,** even for applications with stringent requirements (time critical).
- The proposed computation offloading mechanism is generic and applicable on several types of MEC applications.

Business Impact

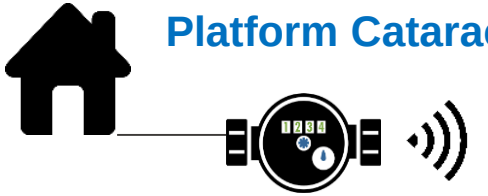
IMPACT ON WINGS BUSINESS

- Lowering cost of servers and penetration of mMTC technology drive the increased use of MEC with combination of IoT devices
- Innovative new services such as Virtual / Augmented reality, UHD video and gaming and real time monitoring and management of critical networks present new business opportunities
- These services present extremely stringent QoS requirements including ultra-low latency, increased energy efficiency and increased flexibility

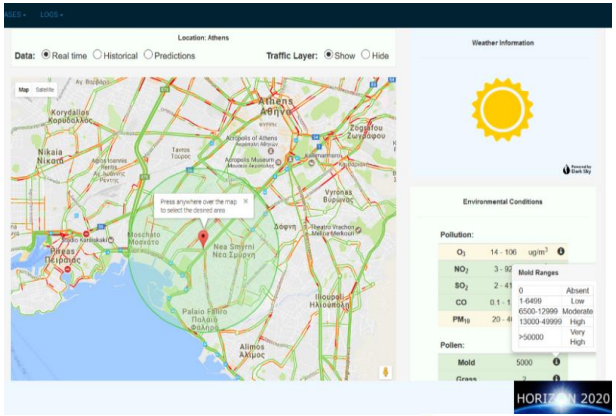
Business Impact

IMPACT ON WINGS BUSINESS

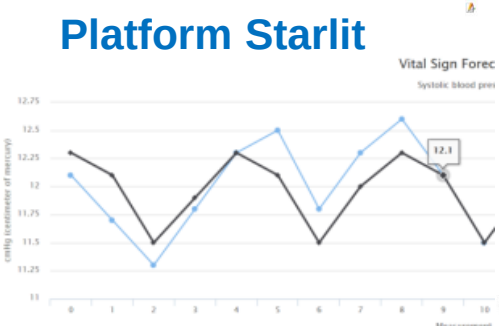
- WINGS has build and operates a number of different IoT platforms providing solutions for water management, smart city, environmental moitoring, etc.
- Enabing MEC functionality with dynamic computational offloading on these platforms could provide substantial business benefits to WINGS

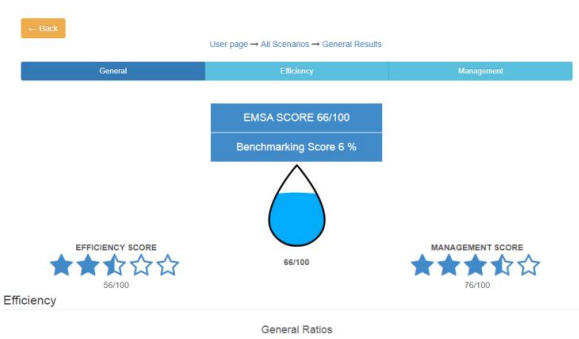


Platform Cataract



Platform Starlit





Business Impact

IMPACT ON WINGS BUSINESS

- Offloading heavy tasks to the edge will result in **providing the necesary latencies for UHD** , virtual & augmented reality applications and real time managements of WINGS platforms
- **IoT devices** and remote sensors will have **extended battery life** and battery / sensor replacement will occur in sparser timelines
- **MEC resources** will be able to be **utilized for other services when not needed for the specific IoT applications** thanks to the dynamicity of the offloading mechanism (reduced server costs)
- Able to **provide the necessary guarantees to stakeholders** (in terms of QoS) which are necessary for new investments in the area.

Feedback

FED4FIRE+ ADDED VALUE

- Lab-developed solutions can be tested in a more realistic environment using large scale deployments
  - Gaining important insights regarding real world applicability of developed solutions
- Funded experiments allow for a more extensive research
- Testing of developed solutions on different platforms offers performance verification and makes debugging / trouble-shooting more efficient
- Interaction with other experimenters offers insights regarding further solution development and cross-domain idea generation

Feedback

- **Pain points during experimentation**
  - No single point experience (with multiple testbeds this is a giant overhead)
    - Initial pre-reservation on the testbed itself needed
    - Later jfed reservation needed
  - No image creation, so the experiment had to be set up from the beginning
  - Significant day to day issues with jfed. Not a friendly tool
  - Low availability.
    - Continuously disconnecting from servers and rebuilding experience. (IMEC)

- **Positive view on**
  - Very good support from testbed stuff, responsive to emails, immediate advice, etc.
  - Availability of very diverse testbeds that are not available anywhere else.


Co-funded by the
European Union


Co-funded by the
Swiss Confederation

This project has received funding from the European Union's Horizon 2020 research and innovation programme, which is co-funded by the European Commission and the Swiss State Secretariat for Education, Research and Innovation, under grant agreement No 732638.

WWW.FED4FIRE.EU