Agenda

- Mobile Edge Computing (MEC)
  - Overview
- Fog Computing
  - Overview
- MEC and Fog
  - State of the art
  - Architectures
  - Functions
- Our proposal: The 5GEE architecture
  - Model
  - Design
- Possible Implementation Blueprint
- Example of Use case
- Conclusions and Future Work
Mobile Edge Computing (MEC)

- The **MEC** architecture is proposed to overcome the challenges of limited-resource mobile devices.

- MEC offers high bandwidth, low latency and support to the mobility of the nodes.

- **Cons**: limited number of edges and low re-configuration rate, due to high costs of configuration and maintenance.
Fog Computing paradigm is proposed to overcome the limits of Cloud Computing

- Fog supports the Internet of Things (IoT) concept
- Fog is more flexible due to its decentralized architecture

- Fog provides data, storage, computing, and application services to end-users thanks to the intermediation of a local proxy, often called Smart Gateway (SG)
- **Cons**: typically fog is used for resource-poor devices and sensing scenarios and SGs are typically unable to host heavy computations
Integration of Fog and MEC paradigms. In the last years there are also a very few seminal works aimed at exploring the idea of integrating Fog and MEC paradigms
  - Merging Fog and MEC functionalities
  - Orchestration of functionalities

Some work is specifically targeted for specific goal and does not provide any in-depth analysis of the pros/cons of integrating MEC and Fog

A general architectural model able to truly ease the integration of the existing MEC/Fog functions is still largely missing
The architectures of MEC and Fog computing share many similarities:

- Central cloud layer
- Middleware (middle) layer
- End devices layer

The crucial **middleware layer**, for both architectures, it provides a set of functions close to the edge.
The MEC server is usually implemented by telco providers to enrich their network infrastructure with new services
  - connected directly at the base station
  - includes telco-oriented facilities

Main MEC functions:
  - MF1: **Execution of resource-intensive applications**. This function enables the execution of telco multimedia and mobile applications close to the edge to grant ultra-low latencies below 1ms (one of the core 5G requirements)
  - MF4: **Caching of multimedia contents**. This function enables content caching, i.e., one of the basic building blocks to minimize round trip time and to maximize throughput for better quality

In our work we identified six main functions for MEC server
The **Fog architecture** is more oriented to industrial Smart Gateways (SGs)

- providing connectivity to the underlying Mobile Heterogeneous Networks (MHNs)
- execute services

**Main SG functions:**

- **FF4**: *Storage and aggregation of data*. This facility allows to store data collected from heterogeneous sensors or devices. Considered the requirements on reliability and durability of data in a given locality, after an agreed period, SG can synchronize the whole (or only the significant part of the) locally aggregated data to the cloud
- **FF7**: *Service Handoff*. This facility allows the transparent service rebinding of end user devices as they move from one (old) SG to another (new) one

In our work we identified **eight** main functions for SG
Architectural solution called 5G-Enabled Edge (5GEE) that aims at converging MEC and fog computing paradigms while maintaining quality awareness and orientation

- **Combination** of all the main MEC and Fog functions and capabilities
- **Dynamic management/(re-)configuration** of 5GEE entities
MEC/Fog functions are similar in existing proposals. In our work, we decide to **smartly combine** functionalities by minimizing overlapping features and by exploiting synergies.

<table>
<thead>
<tr>
<th>Function</th>
<th>Motivation</th>
</tr>
</thead>
<tbody>
<tr>
<td>F2 – Context data service</td>
<td>MEC has network-based information about context, while fog solutions tend to have better information closer to edge</td>
</tr>
<tr>
<td>F7 – D2D communication support</td>
<td>In our solution, the integration of <strong>D2D mechanisms and protocols</strong> is also the key towards locality identification and detection, as well as for enabling localized communications with no load over the MEC/fog-to-cloud trunk</td>
</tr>
</tbody>
</table>
### 5GEE - Model

<table>
<thead>
<tr>
<th>Function</th>
<th>Motivation</th>
</tr>
</thead>
<tbody>
<tr>
<td>F8 – Data storage and aggregation</td>
<td>This facility allows to <strong>store data collected</strong> from heterogeneous sensors or devices</td>
</tr>
<tr>
<td>F10 - Handoff</td>
<td>This is a crucial functionality for our architecture, given our specific interest in the support of mobile IoT applications. In order to guarantee <strong>continuity of service</strong>, the envisioned feature has to exploit predictive mechanisms (also based on profiling), proactive management of involved resources, and live but lightweight migration</td>
</tr>
</tbody>
</table>
IEEE 5GEE - Design

- **Services** and **network functionalities** will be implemented as software components executing on standard operating systems and emerging networking standards
  - Network Function Virtualization (**NFV**)
  - Software Defined Networking (**SDN**)

- In order to manage services and functionalities, we need a **service orchestrator** to take over the deployment complexity across the whole infrastructure
  - **ETSI MANO** (Management and Orchestration)

- We plan to use a particular solution based on the **OpenBaton** MANO framework
  - OpenBaton extends the existing standard specification of ETSI MANO in order to be able to properly manage also MEC applications and to use container deployment tools (e.g. **Docker**).
Three-layer architecture based on the extension of emerging MEC 5G - Fog technologies, by integrating proactive and/or reactive container migration

- **Mobile devices** layer consists of all the endpoints that need to perform high-resource demanding executions of mobile services and do not have enough capabilities to do that
- **5GEE layer**
  - Assist devices to execute computation-intensive tasks and preserve service functionalities in case of user’s mobility
  - Service reactive and proactive handoff migration and fast provisioning
- **Cloud layer**
  - Data storage component
  - 5GEE nodes coordinator through position predictions
Possible Implementation Blueprint

- Possible implementation of 5GEE node
  - Services and functions are containers
  - OpenBaton for orchestrating virtual resources

- **NFV Orchestrator**: responsible for on-boarding of new network services (NS) and virtual network function (NFV) packages

- **VNF Manager**: oversees lifecycle management of NFV instances

- **Virtualized Infrastructure Manager (VIM)**: controls and manages the NFV compute, storage, and network resources
Example of Use Case

- 5GEE may be useful in **Mobile CrowdSensing (MCS)** scenario
  - collect passive and active data (with smartphone) that are processed and shared with other participants and/or with the central cloud
- 5GEE node initially is set to provide telco functionalities
- Alice participates to an **MCS** project. Which support features can help Alice?
  - **F8** (store and aggregate data functionality)
  - **F2** (context data service functionality)
  - and we may also need
    - **F7** (D2D communication)
    - **F10** (handoff)
- Orchestrator dynamically loads the needed functionalities
We propose a **new architectural model** to ease the provisioning and to extend the coverage of traditional edge computing approaches by bringing together the best of MEC and Fog research areas

- Dynamically orchestrate all functions and needed resources at 5GEE nodes
- Docker container solution in order to reduce VMs latency

**Future work**

- Fulfilment of all functions
- Use of 5GEE in Mobile CrowdSensing scenario and, specifically, in the **ParticipAct** living lab
Thanks for your attention!
Questions time...

Contact Info

Domenico Scotece
domenico.scotece@unibo.it