Action full title:

Universal, mobile-centric and opportunistic communications architecture

Action acronym:

UMOBILE

Deliverable:

D5.3 “Proof of Concept (1)”

Project Information:

Project Full Title	Universal, mobile-centric and opportunistic communications architecture
Project Acronym	UMOBILE
Grant agreement number	645124
Call identifier	H2020-ICT-2014-1
Topic	ICT-05-2014 Smart Networks and novel Internet Architectures
Programme	EU Framework Programme for Research and Innovation HORIZON 2020
Project Coordinator	Prof. Vassilis Tsaoussidis, Athena Research Center

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 645124

Deliverable Information:

This deliverable provides the description of the integration of the UMOBILE system as well as the details of the two demos that integrate the proof-of-concept. The proof-of-concept is initially described in this deliverable; then, in M34, the full aspects concerning the proof-of-concept and software availability will be described and reported in D5.4 "Proof-of-Concept (2)".

Deliverable Number-Title	D5.3 Proof of Concept (1)
WP Number	WP5
WP Leader	FON
Task Leader (s)	AFA Systems
Authors	**COPELABS**: Paulo Mendes, Seweryn Dynerowicz, Omar Aponte **ATHENA**: Sotiris Diamantopoulos, Alex Sarros **UCL**: Sergi Rene, Ioannis Psaras **UCAM:** Adisorn Lertsinsrubtavee, Carlos Molina Jimenez **SENCEPTION:** Rute Sofia **TECNALIA**: Iñigo Sedano Perez **AFA**: Angela D'Angelo, Gianmichele Russi, Francesco Amorosa **FON:** Alberto Pineda, Pablo Salvador
Contact	dangelo@afasystems.it
Due date	M24: 31/01/2017
Actual date of submission	M24: 31/01/2017

Dissemination Level:

PU	Public	x
CO	Confidential, only for members of the consortium (including the Commission Services)	
CI	Classified, as referred to in Commission Decision 2001/844/EC	

Document History:

Version	Date	Description

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 645124

Version 0.1	5/11/16	Initial template with Table of Contents
Version 0.2	16/12/16	Revised table of contents
Version 1.0	13/01/17	Revised first version based on feedbacks
Version 2.0	23/01/17	Updated version after plenary meeting
Version 2.1	27/01/17	Added some sections
Version 2.2	30/01/17	Finalized preliminary version of the deliverable, circulated for internal review
Version 2.3	31/01/17	Final version
Version 2.4	04/04/17	Revised version based on comments provided by the reviewers during the interim review

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 645124

Table of Contents

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 645124

List of figures

List of tables

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 645124

Executive Summary

Background

Work Package 5 “**Overall platform integration and validation”** of UMOBILE project aims at the evaluation of the solutions developed in the project. A **proof-of-concept** is expected as an outcome of WP5. Task 5.3 “Proof-of-Concept” concerns, indeed, the integration of the architecture and services, derived from results developed in WP3 and WP4, with different components, such as mobile nodes, sensor nodes, backhaul links of different type, Wi-Fi infrastructure/equipment etc.

This Report is written in the framework of Task 5.3 “Proof of Concept”.

Objectives

The goal of this document is to provide a description of the **integration of UMOBILE system**. The UMOBILE devices are defined; the interaction between software modules is described and will be then evaluated through the proof-of-concept software (please refer to D.5.1 “Validation methodology and evaluation report” for the devices evaluation).

The **proof-of-concept** is created on two specific demos: two independent technological demonstrations based on software developed during UMOBILE. The demonstrations rely on two of the use-cases selected in WP2. The proof-of-concept is initially described in this deliverable; then, in M34, the full aspects concerning the proof-of-concept and software availability will be described and reported in D5.4 “Proof-of-Concept (2)”.

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 645124

1. Introduction

UMOBILE project goal is to develop a mobile-centric service oriented architecture that efficiently delivers content to the end-users, by decoupling services from their origin locations and shifting the host-centric paradigm to a new paradigm that incorporates aspects from both information-centric and opportunistic networking. To achieve this, UMOBILE architecture combines two emerging architecture and connectivity approaches, Information Centric Networking (ICN) and Delay Tolerant Networking (DTN), into one single abstraction.

UMOBILE needs to support various challenged scenarios, such as aftermath of disasters or networks with limited backhaul capacity, that pose several challenges such as increased latency, intermittent connectivity, etc. To address these challenges, UMOBILE architecture includes a resilient service migration module, which utilizes advances in lightweight operating systems to push service instances right to the network edge.

Having already described in detail the envisioned UMOBILE services in D3.3 and the architecture design in D3.1, in this deliverable we focus on the description of the UMOBILE system components from a functional point of view and, mostly, on the integration among the components. While D5.1 provides the evaluation of the single components, the present document focuses on the evaluation of the whole system through the definition of the proof-of-concept.

The proof-of-concept is created based on two independent demonstrations, which rely on two of the use-cases selected in WP2. The proof-of-concept is initially described in this deliverable; then, in D5.4, the full aspects concerning the proof-of-concept and software availability will be described.

In addition to the aforementioned, the present deliverable includes also a short guideline to the usage of the UMOBILE Lab, which will be used as testbed for the proof-of-concept, as well as details on the implementation of the Lab (attached to D5.3 as Annex A).

This document is organized as follows.

- **Section 2** provides the description of the integration of UMOBILE system. The UMOBILE devices are defined and the interaction between software modules is described.
- **Section 3** describes the UMOBILE Lab, the testbed that will be used for the proofs of concept described in Section 4.
- Two different proofs of concept, corresponding to two of the use-cases selected in WP2, are reported and analyzed in **Section 4**.
- Finally, in **Section 5** we draw the conclusions.

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 645124

2. UMOBILE System Integration

Figure 1 (extracted from D3.3) shows the high-level design of the UMOBILE architecture divided into two distinct domains: the **UMOBILE domain** and the **Internet domain**.

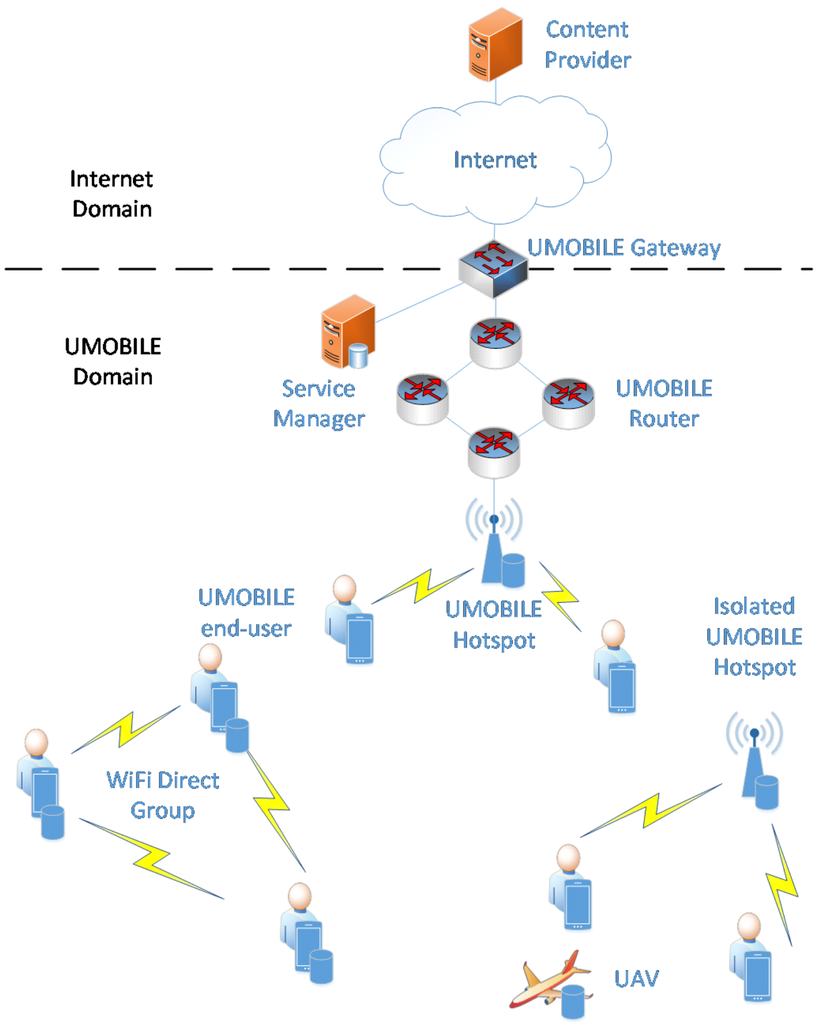


Figure 1. Overview of the UMOBILE platform

The actors involved in the UMOBILE architecture, as already shown in previous deliverables, include:

- **UMOBILE-enabled end-user devices** (i.e., smartphone, tablet), used to send and receive participatory data (e.g., photos, short messages) as well as opportunistic data (e.g., atmospheric pressure, temperature, noise, roaming patterns).
- **UMOBILE-enabled hotspots** are able to collect and relay relevant information (e.g., alert messages, instructions from emergency authorities), host some instantiated services or store collected data, check its validity and perform computational functions (e.g. data fusion) to increase the value of the information to the civil authorities.

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 645124

- **UMOBILE-enabled UAV devices** able to collect and relay relevant information and connect two isolated areas.
- **UMOBILE-enabled gateways/proxies** provide interconnectivity between UMOBILE domain and the Internet domain.

UMOBILE is being developed as a modular software architecture, where some modules may or may not reside in a specific hardware element. From a functional point of view, we can further classify the UMOBILE architecture in the following way:

- UMOBILE GATEWAY (1)
- UMOBILE SERVICE MANAGER (2)
- UMOBILE HOTSPOT (4)
- UMOBILE End-User SERVICES (3), which include:
    - Some services running in background (users unaware)
        - Contextual manager
        - NDN-Opp
        - KEBAPP
        - NREP
    - A List of APPs (users' aware, based on their preferences)
        - Oi!
        - Now@
        - Route-Planner

Based on the previous classification, the UMOBILE architecture can be further described as in Figure 2, which is a broadening of Figure 1 as it contains different scenarios where the UMOBILE platform can find application. Figure 2 highlights the relationship among the defined components.

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 645124

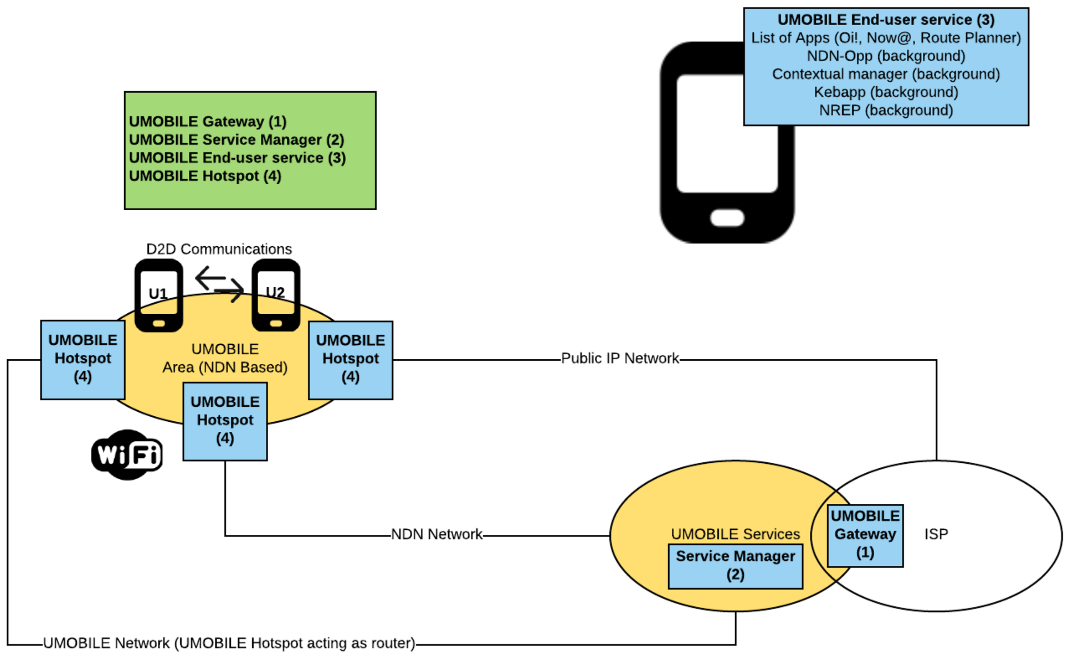


Figure 2. Functional blocks of the UMOBILE platform

The following sections describe the defined functional components focusing on the interaction among modules. Please refer to D.5.1 “Validation methodology and evaluation report” for a detailed description and validation of each of the components.

2.1. UMOBILE Gateway

The UMOBILE Gateway provides interconnectivity between the UMOBILE domain and the Internet domain. Such device can be employed by service and content providers to act as repositories, being able to store data received through the IP network and then to share it over the UMOBILE network (or vice-versa) upon request. In the framework of the UMOBILE project, focus is on service and content sharing over the UMOBILE network.

UMOBILE Gateways can be part of the infrastructure of service/content providers or civil authorities, being employed as repositories supporting both the IP and the UMOBILE part of the network. In particular, providers can utilize UMOBILE Gateways as “entry point” of the UMOBILE network, connecting their legacy IP-based infrastructure with other UMOBILE nodes, as depicted in Figure 2. This point of entry is normally expected to reside in the provider’s premises. Depending on the deployment model, though, any UMOBILE hotspot can act as Gateway, if it is equipped with an IP interface and has sufficient storage and processing power.

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 645124

The gateway assumes its role by providing universal access to content or services located in both the host-centric and information-centric domain. For example, a service in the IP domain can be "fetched" by the UMOBILE gateway – with the help of the Service Manager – and stored in the UMOBILE part of the network. Thus, it is available to the UMOBILE users by the repository; the users can choose to download it and deploy it, as they like. In essence, the gateway provides a common pool of shared services and information between the two domains, assisted by the service migration mechanisms.

As a UMOBILE node itself, a UMOBILE Gateway is employing the full UMOBILE platform, being, thus, able to forward data over the DTN interface if required.

2.2. Service manager

A central aim of the UMOBILE project is to build a service-centric architecture that is capable of supporting the deployment of a diverse set of services with different QoS requirements ranging from best—effort to guaranteed QoS with different degrees of stringency. We address the challenge by means of integrating the abstractions natively provided by the ICN paradigm, Delay Tolerant Network (DTN) techniques and opportunistic service migration to the edge of the network. In pursuit of this aim, we have developed three technologies:

- DTN framework,
- Service Migration Platform,
- KEBAPP (Keyword-Based Mobile Application Sharing).

The details of these three technologies are described in D3.3 "UMOBILE ICN layer abstraction initial specification"; we now focus on the Service Manager, which is one the devices we can identify in the UMOBILE architecture, as depicted Figure 2.
A detailed discussion of service migration, and the role of service manager in the UMOBILE architecture, is presented in D3.1 "UMOBILE architecture report (1)". In this section, we present a high-level overview about its functionality, focusing on its integration with DTN and KEBAPP.

2.2.1. Network architecture integration of service migration, DTN and KEBAPP

For the sake of clarity, in order to show the potential exploitation of Service Manager in practical applications, we will use the hypothetical emergency scenario discussed below. It is worth emphasizing that the scenario is a simplified version of the use case documented in D2.1 "End user Requirements Report". The same scenario will be then included in the proof-of-concept described in section 4.

1. *Imagine an area that has been struck by an undesirable event such as a fire, floods, or earthquake, etc.*

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 645124

2. *Consequently the network infrastructure in that area is temporarily disturbed by the event.*
3. *Upon being notified (over a secondary channel) about the situation, a rescue team arrives to the area and contacts (over a secondary channel) the Service Manager to request the deployment of emergency computing services in the affected area.*
4. *The expectation is that the services are available for the benefit of the rescue team and the victims of the event under the observance of specific QoS requirements. For instance, some services might tolerate arbitrarily long and unpredictable latency whereas others are latency-sensitive. Examples of such services are web servers that offer news and maps of affected area.*

We now explore how the request of the rescue team can be fulfilled by the combined efforts of the service migration platform, DTN framework and KEBAPP framework. We open the discussion with a description of the network infrastructure and proceed to explain the integration of the service migration platform and DTN and service migration platform and KEBAPP. The integration of the three technologies together is left for future work.

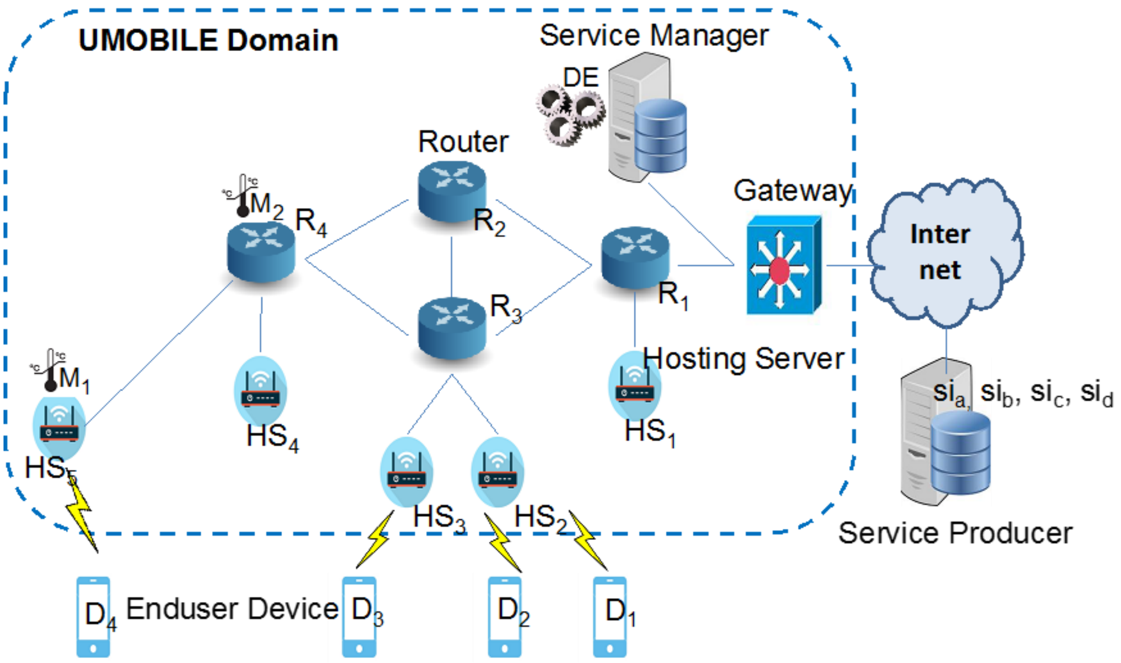


Figure 3. A view of UMOBILE service integration (with focus on service migration)

Figure 3 shows the core of the network architecture for the integration of the service migration platform, DTN and KEBAPP.

- **UMOBILE Domain**: The UMOBILE Domain is a set of nodes deployed with UMOBILE software so that they are able to take advantage of the NDN facilities. It includes network routers (for example, R3) and applications hosts (for example, HS2). All the components shown in the UMOBILE domain require the UMOBILE platform software to support applications such as these demonstration scenarios. However,

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 645124

the deployment of UMOBILE platform software in end users' devices is optional since these devices can use conventional IP interfaces to connect to the UMOBILE domain. A UMOBILE Gateway links the UMOBILE Domain to the conventional Internet.

- **Gateway**: The UMOBILE Gateway is responsible for connecting the UMOBILE Domain to the Global Internet. Its functionality is to convert NDN Interest Requests to HTTP requests and HTTP responses to NDN Response. In the figure, it is deployed in one of the NDN routers (R0).
- **Routers** (R1, R2, R3, R4): standard NDN router. We assume that UMOBILE Routers are in possession of storage that they use for in-network caching and for storing application level information such as *dockerized* service images.
- **Hotspots** (HS1, HS2, HS3, HS4): A Hotspot is a conventional computer with wireless communication facilities that can offer connectivity to End-user Devices (D1, ..., D4). It has disk storage facilities and virtualization software. In our experiments, we use Docker virtualization technology. Likewise, to implement the Hotspots we use Raspberry Pi computers that are capable of executing Linux containers.
- **End-users devices** (D1, D2, D3, D4): An End-user device is a mobile device with wireless facilities and interested in accessing services provided by the ISP provider. We assume that mobile devices communicate with the UMOBILE Hotspots over conventional HTTP.
- **Service Provider**: We assume the emerging business model where network providers are responsible for providing both network connectivity and access to services to end-users. In the figure, we assume that the service provider is the owner and in full control of the resources included in the UMOBILE Domain. Consequently, the Service Producer has delegated to the Service Provider the responsibility of deploying the services.
- **Service Producer**: It is an entity in possession of some arbitrary services of interest to the end-users. He stores them as compressed *dockerized* images si_a, si_b, si_c and si_d which are at the disposition of the Service Provider.
- **Services images** (si_a, si_b, si_c, si_d): A service image is a compressed *dockerized* images stored within the Service Producer.
- **Services** (s_a, s_b, s_c, s_d): A service is an application of interest to the end-user that can be instantiated from a corresponding compressed *dockerized* image. We assume that the services demand different levels of QoS, for instance different latencies. For example, service s_a is latency-sensitive whereas s_d is latency-tolerant.
- **Service Manager**: The Service Manager is a piece of software that implements all the functionality that the Service Provider needs to deploy his services, including disk space to store both services and compressed images. In the figure, the Service Manager is strategically deployed on a computer directly connected to the Gateway. This deployment simplifies the task of transferring compressed service images from the global Internet to the UMOBILE Domain. At the heart of the Service Manager is a

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 645124

decision engine that is responsible for the actual deployment of services.

- **Decision Engine** (DE): The decision engine is a piece of software with all the necessary logics to make decisions about service deployment and migration. The decision engine has access to monitors deployed strategically within the UMOBILE Domain and instrumented to collect metrics about conditions of interest. On this basis, it decides about which services to deploy, when and where. For example, it creates a second replica of a given service in a neighbor Hotspot when the monitoring reports that the first replica is overloaded.
- **Monitors** (M1, M2): A monitor is a software tool for collecting real time information about the status of the resources included in the UMOBILE Domain, such as network conditions and current usage of their Hotspot resources (CPU, memory and disk). Only two monitors are shown in the figure, yet there is nothing to prevent from deploying as many as necessary. A good example of a monitor is a Python script deployed in the Raspberry Pi used to implement a Hotspot to measure and report its current free memory.

In subsequent sections, we explain how the core infrastructure can be used for addressing the QoS requirements demanded by the emergency scenario discussed above.

2.2.2. Integration of service migration with DTN

This section explains how the integration of the service migration platform with the DTN framework can help to deploy services in a location with partitioned from the main network. The aim is to demonstrate how the service migration platform can be integrated with the DTN framework to mask network problems (for example connectivity failure) to honor QoS, such as those demanded by the described scenario.

We will explain a potential situation with the help of Figure 4, which is the result of an adaptation of Figure 3.

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 645124

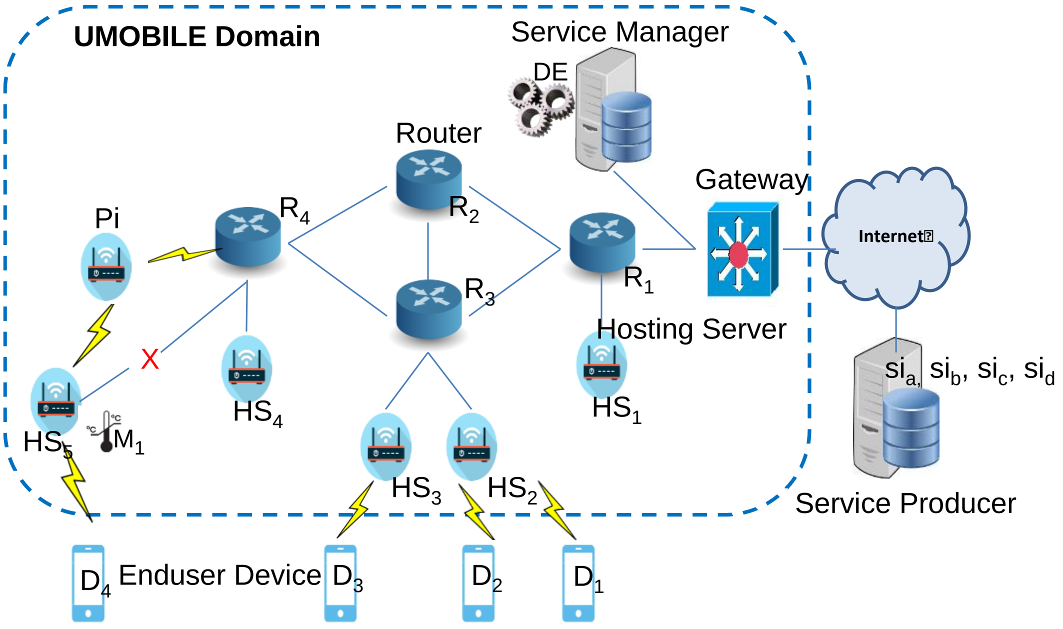


Figure 4. A view of UMOBILE deployment with focus on service migration

2.2.2.1. Service deployment request

1. Let us assume that HS5 is located is the affected area and consequently, as shown in the figure, the network cable between HS5 and R4 is broken down. We assume that the routers and the Hotspots are instrumented with the service migration and DTN software that we have developed, such as the Pi (a conventional Raspberry Pi computer) shown in the figure. The Pi is only a specific realization of a mobile device with wireless facilities. It can equally be realized as a conventional Android mobile phone or by a UAV capable of flying over the affected area (backwards and forwards between R4 to HS5). The monitor M1 attached to HS5 is a Python script that provides the Decision Engine with the status of the CPU, memory and disk HS5. We will deploy a monitor in each Hotspot and a similar tool in strategic locations to measure network conditions.
2. Upon arrival to the area, members of the rescue teams use their mobile devices (for example, mobile phones) to connect to the Hotspots and request the Content Provider to deploy emergency services s_a and s_d in HS5.
3. Let us assume that the Service Manager is aware that the functionality of s_d is not affected by the latencies and that s_a is. On this basis, the Service manager configures his decision engine (not shown in the figure) to:
    a. deploy s_d in the core of his network (rather than attempting to deploy it as close as possible to the end user). sd will be deployed in any of the Hostspots as long as the selected one has enough CPU, memory and disk resources to

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 645124

host it. This point will demonstrate that the Decision Engine is aware of the different level of QoS that the services need.

b. deploy s_a in HS5 using the facilities provided by the DTN. This is a sensible solution provided that s_a can operate independently, that is without the support of the core ICN network infrastructure. To achieve this task, the Service Manager copies the s_{ia} to R4 and instructs the Pi to upload it and download it to HS5. This point will demonstrate the functional integration of the service migration platform and the DTN framework. We will quantify the results as explained next

2.2.2.2. Service migration facilities

To copy si_a from the Service Producer repository to R4, the Service Manager uses the facilities of the service migration platform that we are implementing based on abstractions offered by the NDN network. Central to the execution of this task is the Push-based communication model with publish data dissemination explained in D3.3 UMOBILE ICN layer abstraction initial specification. In brief, in this model, the content producer (the Service Manager in this example) initiates the data transference to the data consumer (to R4 in this example.).

1. The Service Manager issues an Interest packet (for example, "r4.com/services/emergency/nolatency_tolerance/sia/pub") for the benefit of R4 who has registered an interest with the Service Manager.
2. Upon receiving the notification, R4 issues an Interest packet (for example "servicemanager.com/services/emergency/nolatency_tolerance/sia) against the Service Manager to fetch the sia.

The responsibility of deciding what services to deploy, when and where lies with the decision engine. The decision engine is instrumented with the necessary logics to make decisions, for example taking this simple scenario and assuming that s_d and s_a are of type 1 and 2 respectively, the decision engine can be programmed to execute:

```
switch (serviceType){
case 1:  /* latency tolerant services */
          if (local HostingServer found) then triggerLocalDeployment
          else send Error to Service Manager;
          break;
case 2:  /* latency sensitive services */
           if (remote Hotspot found) then trigerRemoteDeployment
          else send Error to Service Manager;
          break;
}
```

Such a logic can be implemented as a conventional Python script code or alternatively, as

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 645124

declarative rules to be reasoned about and executed by a reasoning engine like Intellect, PyKE, Drools or PyCLIPS, all depending on the complexity of the decisions to be taken.

We can describe the operation flowchart in the following way:

1. Deploy monitors (Python or shell scripts) in strategic locations to collect metrics about the status of the Hotspots and their network links and report their observances to the decision engine.
2. Use tools (Python or shell scripts) to place requests against the Service Manager to deploy s_a and s_d.
3. Use tools (Python or shell scripts) to place requests against s_a running in HS5.
4. As point 3 is executed, use tools (Python or shell scripts) to collect D4-HS5 latency metrics and plot them. They should be or of the order of a few milliseconds.

For the sake of comparison, and knowing that HS1 has enough resources to host s_a we will use it to host s_a and conduct the following experiment to demonstrate that s_a is not operational under poor latency guarantees.

1 Use tools (Python or shell scripts) to place requests against s_a hosted in HS1, measure D4-HS1 latency and plot the results.

2 The results will explain that the deployment of s_a in HS1 is unsuitable.

2.2.3. Integration of service migration with KEBAPP

We devise two scenarios where the KEBAPP service and the service migration platform can work integrated providing interesting features to the project. In the first case, the service migration platform migrates a local service to the hotspots. Once migrated, the service can be accessed and shared among users through KEBAPP-enabled applications. In the second case, the service migration can migrate a local repository to the UMOBILE hotspots in order to provide to the users that are not KEBAPP-enabled, the apk (the application installer of the KEBAPP-enabled application) necessary to enable local communications with other users.

KEBAPP-enabled services in UMOBILE hotspots

KEBAPP is an application-centric framework for opportunistic computing at the edge of the network on mobile devices such as smartphones and tablets. It allows a mobile device to exchange information, in an opportunistic way, using smartphone apps, locally, with other devices that happen to be in that area. By locally we mean without involving communication with the global Internet. Examples of information that can be exchanged by KEBAPP-enabled devices are traffic problems, shopping opportunities in the area, lost-and-found articles, etc.

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 645124

To be able to participate in a KEBAPP group, a mobile device needs have installed the KEBAPP framework included in the UMOBILE end-user service (presented as an apk file) and a KEBAPP-enabled application.

A KEBAPP group can be a set of users sharing the same application that want to locally communicate to other users to share content and/or computation resources. For example, a KEBAPP-enabled application can be a Route-Planner application, able to calculate routes for other users that do not have the information necessary to calculate those routes or do not have connection to Internet. Another example could be a concert scenario, where a group of users can share pictures of the event using the same application. However, in all these examples, some problems may arise when a set of users are connected using Wi-Fi Direct without any participation of the infrastructure, such as the following.

- Battery: Wi-Fi Direct communications can waste smartphones battery, especially for the group owner (the leader that manages the Wi-Fi Direct group).
- Intermittent connectivity: Users' mobility can generate several disconnections, and therefore instability in the communications or broken messages, especially when the group owner leaves the group that implies a disconnection of the entire group.
- Limited resources: Caching and processing capabilities in smartphones are more than enough; however, users may rather avoid sharing smartphone resources with other users, such as local storage, data plans or other resources.

For all these reasons, we think deploying KEBAPP-enabled services in UMOBILE hotspots can have benefits, in order to centralize communications between users without resource limitations and providing more stable communications

UMOBILE store migration

To be able to participate in a KEBAPP group, a mobile device needs to download and deploy the KEBAPP-enabled application (presented as an apk file) from a conventional Apache web server (for example, Google play). Let us call it the *UMOBILE store* (UStore).

The drawback of downloading KEBAPP apk file from the core of the network is the traffic cost and the time to download. In this section, we will demonstrate that the aforementioned problems can be avoided by the opportunistic deployment of the UMOBILE store as close as possible to where the potential KEBAPP users are. The central idea is to regard the UMOBILE store as a service that can be *dockerized* and manipulated (deployed, replicated, etc.) by the service migration platform.

The network infrastructure that we will use for integrating the service migration platform and KEBAPP is shown in Figure 5. UStore represents the UMOBILE store whereas KEBAPP

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 645124

group is a group of users that can potentially run KEBAPP.

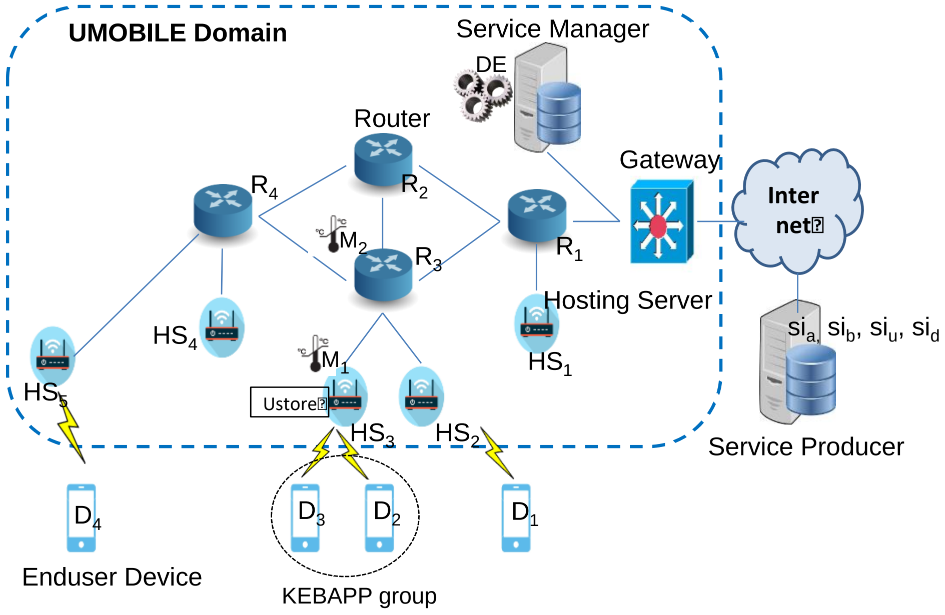


Figure 5. Integration of service migration and KEBAPP

2.2.3.1. Service migration facilities

Let us assume that the provider of the ICN infrastructure of Figure 5 has been delegated the responsibility of deploying a KEBAPP-enabled service (we consider a UMOBILE store as a KEBAPP-enabled service as well) and imagine the following situation.

1. The KEBAPP-enabled service has been *dockerized* and stored as a compressed image, say si_K in the Service Producer from where the Service manager can retrieve it when needed.
2. D2 issues a request against the Service Manager to expressing interest in the deployment of KEBAPP. Let us assume that D2 has access to HS3, HS2 and HS1.
3. Upon receiving the request, the Service Manager instructs his Decision Engine to deploy si_K as close as possible to D2 but under the consideration of the current network conditions, the features (disk, memory and CPU capacity) of the potential Hotspots (HS3, HS2 and HS1) and their current status such as currently available memory. The idea is that some of the candidate Hotspots might not have the resources to run the KEBAPP-enabled service.
4. To make informative decisions, the decision engine relies on up to date information collected from several sources such as network monitors and actual Hotspots. Of central interest are parameters related to QoS. For instance, a Hotspot will be selected by the decision engine when certain conditions hold, for example, when Round Trip Time is above the acceptable level, or memory usage is above 85%.

2.2.3.2. Service deployment technology

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 645124

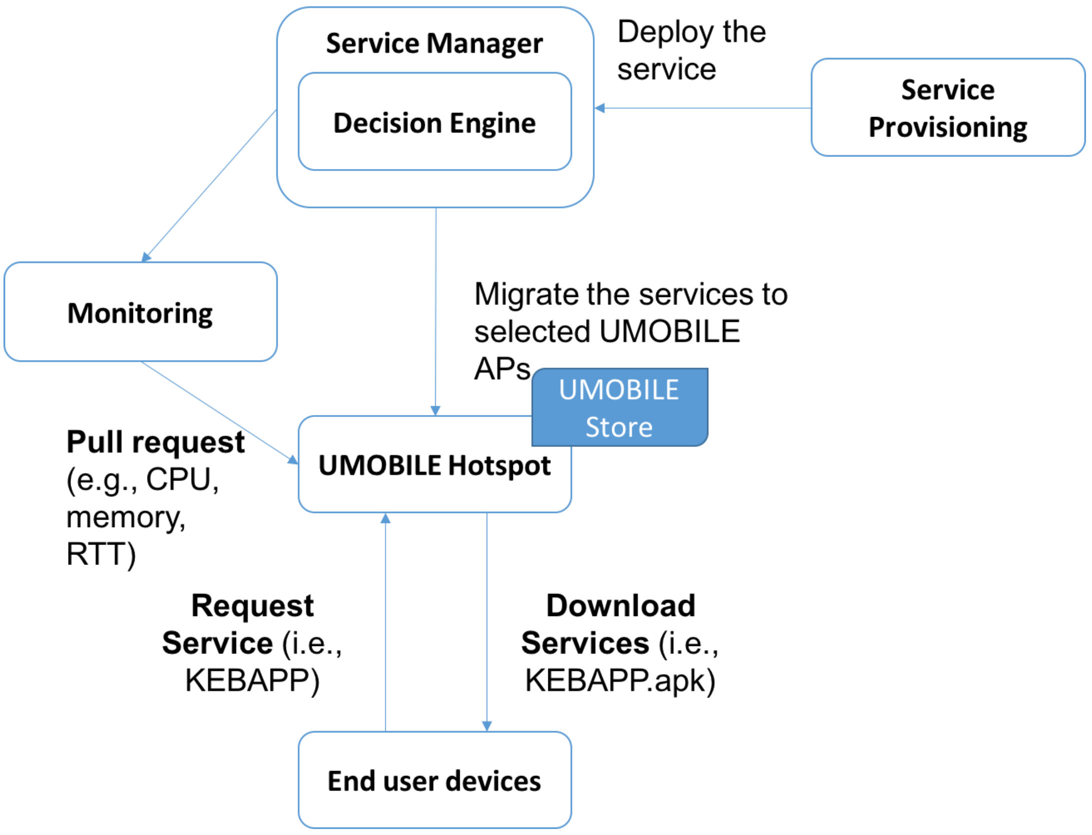
The decision engine collects memory usage data by means of pull requests. Upon a successful migration, the selected UMOBILE hotspots are able to serve the KEBAPP-enabled service locally. This service migration process will run over NDN while utilizing both push and pull communication models.

The aim is to demonstrate how the service migration platform can be used to deploy Ustore at the edge of the UMOBILE platform, for example in HS3, HS2 or HS1 on the basis of the reports sent to the decision engine by the resource consumption monitors deployed in HS3, HS2 or HS1. We can describe the operation flowchart in the following way:

1. Deploy monitors (Python or shell scripts) in HS3, HS2 and HS1 to collect metrics about the status of their CPU memory and disk resources. Measured metrics are sent to the decision engine.
2. Use tools (Python or shell scripts) to enable D3 to place requests against the Service Manager to deploy Ustore and to download the apk KEBAPP file as soon as it becomes available. D3 awaits for a response.
3. The decision engine will deploy Ustore in HS3 as long as HS3 has enough resources to host the Ustore service.
4. D3 request should be satisfied.
5. Use tools (Python or shell scripts) to force the CPU, memory and disk consumption of HS1, HS2 and HS3. For example, bring the resources of HS1 and HS2 to exhaustion and instruct D3 to place requests as in point 2.
6. Verify that the decision engine is capable of deploying Ustore in a Hotspot with enough resources. For example, the decision engine never selects a Hotspot with 75% of its memory consumed.

Figure 6 presents the operation flowchart of the integration of service migration and KEBAPP. The service-provisioning block operated as a connection point between the service migration framework and service/content providers (e.g., rescue team, fire fighters) that want to deploy their services on a UMOBILE network such as the infrastructure shown in Figure 4. In this example, we will deploy the UMOBILE KEBAPP-enabled service to demonstrate the feasibility using KEBAPP in the emergency and civil protection scenarios. The decision engine inside the service manager function block collects information from a monitoring system and feeds it to a rule-based reasoning engine that makes decisions about where and when to migrate a service (the UMOBILE store in this example). The monitoring system fetches information about network condition and available resources (e.g., CPU, memory) from all UMOBILE APs through a push and pull communication model (details discussed in D3.1 and D3.3). After migrating the UMOBILE store service, the end users can use their mobile devices to connect to the UMOBILE Hotspot to download the KEBAPP apk file.

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 645124



Service Manager
Decision Engine
Deploy the service
Service Provisioning
Monitoring
Migrate the services to selected UMOBILE APs
UMOBILE Store
Pull request (e.g., CPU, memory, RTT)
UMOBILE Hotspot
Request Service (i.e., KEBAPP)
Download Services (i.e., KEBAPP.apk)
End user devices

Figure 6. Flow chart of service migration with KEBAPP application

2.3. UMOBILE Hotspot

Typically, wireless (or Wi-Fi) hotspots are essentially wireless access points providing network and/or Internet access to mobile end-user devices, e.g., in public locations. UMOBILE-enabled hotspots are specific UMOBILE network devices, like usual hotspots, but they support the UMOBILE architecture, can run services locally and are compatible with the UMOBILE services.

UMOBILE-enabled hotspots may be able to collect and relay relevant information (e.g., alert messages, instructions from emergency authorities), host some instantiated services or store collected data, check its validity and perform computational functions (e.g., data fusion) to increase the value of the information to the civil authorities.

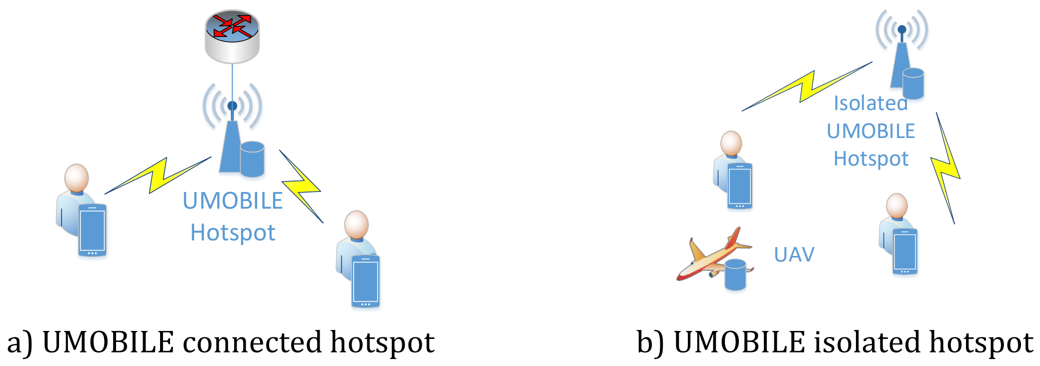


Figure 7. UMOBILE hotspot

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 645124

The specific characteristics of the UMOBILE-enabled hotspot (UMOBILE AP) are the following:

- UMOBILE hotspots can be isolated (not connected to the Internet) or connected to Internet.
- UMOBILE hotspots can host local services deployed using the service migration platform.
- UMOBILE hotspots can provide KEBAPP-enabled services that UMOBILE users are not able to provide due to network restrictions, data-plan restrictions or battery restrictions.
- UMOBILE hotspots can support DTN, through IBR-DTN, and can integrate service migration.
- UMOBILE hotspots can be deployed in isolated UAVs, providing local services and DTN capabilities, or in connected UAVs (through a data uplink), providing connectivity services.
- UMOBILE hotspots can be compatible with NREP in order to prioritize emergency services and messages.
- UMOBILE hotspots can gather social and context information that will be later used by the NDN-Opp and/or PerSense.
- UMOBILE hotspots can be collocated with gateway functionalities in order to provide IP network services to the local opportunistic network. E.g. through KEBAPP Services (e.g. map service) to UMOBILE network (gateway function).

2.4. UMOBILE End User Services

For UMOBILE deployment, the users need the UMOBILE end-user Services (UES) installed. The UES comprise a set of UMOBILE services running in background (of which users are unaware) and a list of native applications, which can be used to experience the performance of UMOBILE under different settings.

The **NDN-OPP** is a new branch the NDN Forwarding Daemon, currently being developed to the Android platform with extensions to function according to an opportunistic networking paradigm. These additions include the implementation of new forwarding strategies, along with support for Wi-Fi Direct communications.

The **UMOBILE Contextual Manager** is a UMOBILE service that runs in background, and captures information concerning the device affinity network (roaming patterns and peers over time and space) as well as concerning usage habits and interests (internal device information). Metrics derived from such contextualization are then passed, upon demand or periodically, to other UMOBILE modules (e.g. to the Routing module, where NDN-Opp

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 645124

resides) to assist in different network operational aspects.

Another service running in background is **KEBAPP** that enables users to share their own applications with nearby users. In a sense, the client application instance can also act as a server instance in order to serve requests from nearby users.

NREP service is a name-based replication priority mechanism, which considers prioritization rules to spread emergency information, along with indicators of the social behavior of users collected via the UMOBILE Contextual Manager Module, to assist in a more efficient data dissemination.

Some applications exploiting UMOBILE benefits are: Oi!, Now@ and Route-Planner.

Now@ is an Android application, which enables users to share data, based on their interests over an NDN infrastructure**.** It is based on a distributed implementation, which requires the client apps to include a synchronization scheme for data pertaining to each interest.

In what concerns **Oi!**, this is an application which allows the users to exchange messages independently of the availability of Internet access, by exploiting the direct wireless communications capabilities (i.e., Bluetooth and Wi-Fi direct) available in personal mobile devices (please refer to D.3.1 for details).

Route-Planner is a KEBAPP-enabled application which allow users request route calculations to other users nearby that are using the same application without connecting to the Internet or without having the maps locally

The end-user devices shall also have installed **PerSense Mobile Light**, which is an example of an external application that the Contextual Manager can be plugged to, to collect data. The purpose is to show how the Contextual Manager can be easily extended to collect new parameters, by relying on external applications.

The end-user UMOBILE services are being designed in a modular way, expected to be cross-compilable for a multitude of platforms, and taking into consideration the potential for embedded systems. Nevertheless, validation will be done in Android devices and, if feasible, in UNIX devices also.

2.4.1. Operation flow chart

Figure 8 provides a simplified illustration of the basic implementation of a UMOBILE end-user device. The UMOBILE software package is composed of NDN-Opp along with services

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 645124

running in background (Contextual manager, Kebapp, NREP) on top of which user applications can be built (e.g.: Oi!, Now@, Route-Planner).

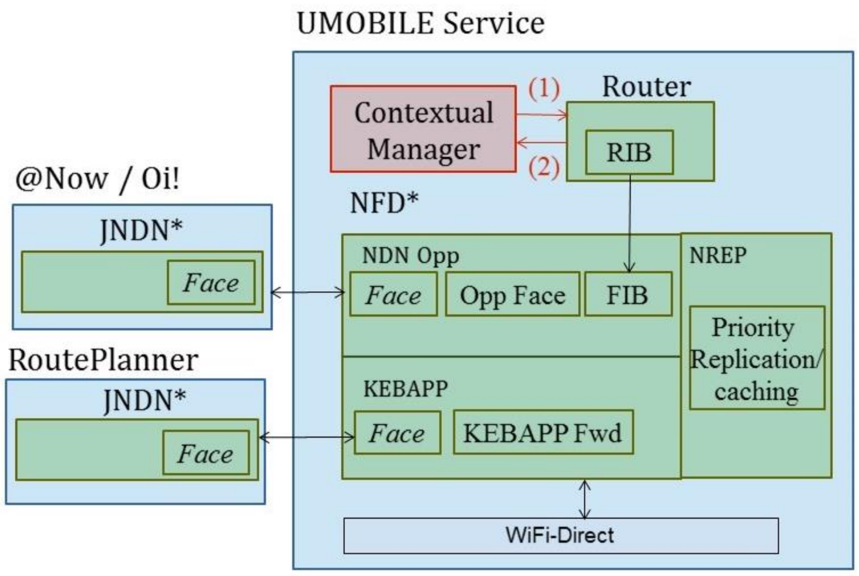


Figure 8. UMOBILE End-user device

The architecture of a UMOBILE end-user device includes the NDN-Opp module, the KEBAPP framework and the NREP system, which packages a modified version of NFD along with the Router, the Contextual Manager and various applications that make use of them for communication. The interface between NDN-Opp and the Contextual Manager is composed of two means of communication:

- Operation (1) is a request from NDN-Opp to the Contextual Manager to get a collection of values of INDICATORS concerning DEVICES within a certain TIMEPERIOD. This information is used to re-compute the routes.
- Operation (2) is a notification by the Contextual Manager to the Router regarding changes of connectivity in the set of devices nearby. Specifically, it is envisioned as providing the Router with a list of devices along with the change to their availability (devices can become AVAILABLE or UNAVAILABLE).

NDN-Opp and KEBAPP are two complementary mechanisms integrated into a single end-user system and developed over the NFD daemon. Both functionalities aim at different purposes, NDN-Opp is aimed at social routing and KEBAPP is aimed at one-hop application sharing. Therefore, both software are developed in the same platform but they do not share specific interfaces (others than the ones already available in the NDN implementation) since are independent. NREP is the third component, and will not be a communication system by itself, but a mechanism to enable priorities for emergency services based on different parameters, such as application priority, time validity or spatial scope. NREP

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 645124

could be integrated with both, NDN-Opp and KEBAPP. In the next version of this deliverable (D5.4) this integration will be reported.
KEBAPP is being developed in WP3 and the details of its operation and implementation will be described in D3.1 and D3.2. NREP is being developed in WP4, the details of NREP mechanism will be reported in D4.3.

We now provide an initial description of the NDN-Opp functionality, starting by describing the Router, as illustrated in Figure 9.

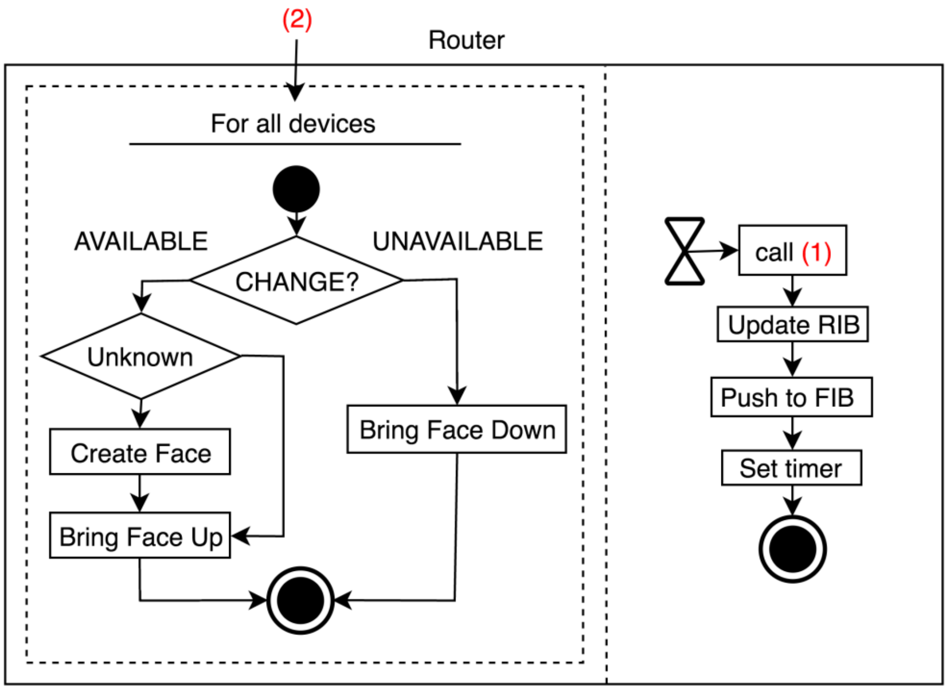


Figure 9. Router in a UMOBILE End-user device

The Router essentially fulfills two functions.

On a periodic basis, it performs the Routing function that is decomposed as follows.
First, the Router uses interface (1) in order to obtain a fresh set of INDICATORS. Second, it performs an update to the Routing Information Base (RIB), which consists in computing the value of the metric for the various routes stored therein. Third, the Router performs the selection and introduction of the routes into the Forwarding Information Base (FIB) of NFD*.

The second function (Face management) is performed in response to a notification (2) by the Contextual Manager of a list of changes to the availability of neighboring devices. The Router processes each entry of the provided list as follows. If a device becomes UNAVAILABLE, the associated OppFace in NFD* is brought DOWN. On the other hand, if a device has become AVAILABLE, its OppFace must be brought UP in response. However, in the event the device was previously unknown, a new OppFace must be created for it first.

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 645124

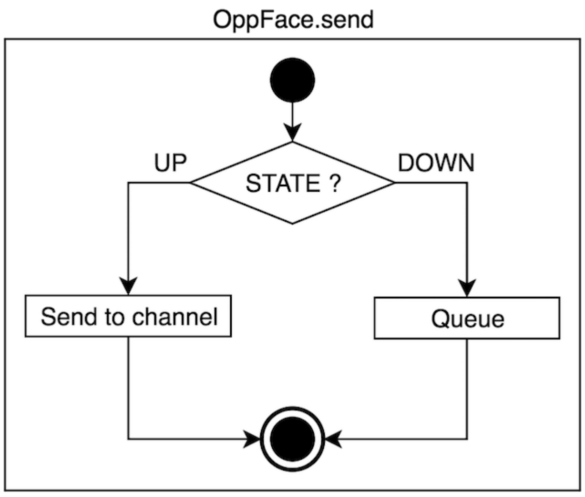


Figure 10. OppFace of NDN-Opp

The opportunistic nature of the NDN-Opp includes the need to have a CARRIER function introduced into the node. The new OppFace introduces a queuing mechanism that implements this function. Whenever the forwarding logic of NFD* decides to send a packet (Interest, Data or Nack) down an OppFace, the behavior depends on its STATE. If it is UP (i.e., communication is possible), the packet is transmitted over the channel to the next-hop. However if the OppFace is DOWN (i.e., communication is not possible at the moment), the packet is placed into a QUEUE until the next-hop becomes available.

When an OppFace is brought UP or DOWN, its STATE must be set accordingly. However, in the case it is brought UP, the packets in its QUEUE can be transmitted over the channel to the next-hop.

The modifications introduced in NFD (viz. packet format) must be reflected into a Java library which is used by the Applications to communicate over NDN.

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 645124

3. UMOBILE Lab

3.1. Description

The lab is conceived to be remotely used by each partner through a VPN connection and is constituted by a number of devices as Wi-Fi access points, Linux systems (Raspberry PI) and Android systems (Banana PI). The lab is finalized to cover the proposed UMOBILE use cases. It allows test sessions in which the human activity is simulated through the Android devices and the results are automatically collected.

The lab is intended to proof the integration of the architecture and services, developed in WP3 and WP4 with different components, such as mobile nodes, sensor nodes, backhaul links of different type, Wi-Fi infrastructure/equipment. It aims to become a complete prototype implementation of UMOBILE platform and the proof-of-concept, which integrates the selected modules, derived from results developed in WP2, WP3, and as WP4. Specific applications developed in the course of the project will be initially demonstrated in the lab, with the software modules developed during UMOBILE. Whenever feasible, and based on the dissemination plans of WP6, the lab shall be used to provide project results, and even to collect data in different events (e.g. conferences), data which can then be provided to the community to enhance further studies.

In accordance with the objectives of WP5, the lab will help to:

- practically demonstrate the overall platform of the project;
- quantitatively evaluate the outcomes of WP3 and WP4;
- implement scenarios with increasing complexity;
- improve the performance of the operational procedures;
- test the limits of the system and of its operational capabilities.

At the end of the project, the lab will be an **overall technological validation of UMOBILE** platform, including a **working proof-of-concept**.

Figure 12 outlines a testbed, with the UMOBILE gateway, some UMOBILE hotspots (access points), and some UMOBILE users. In a more realistic testbed, the APs will be far away from the wired network, as in Figure 13.

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 645124

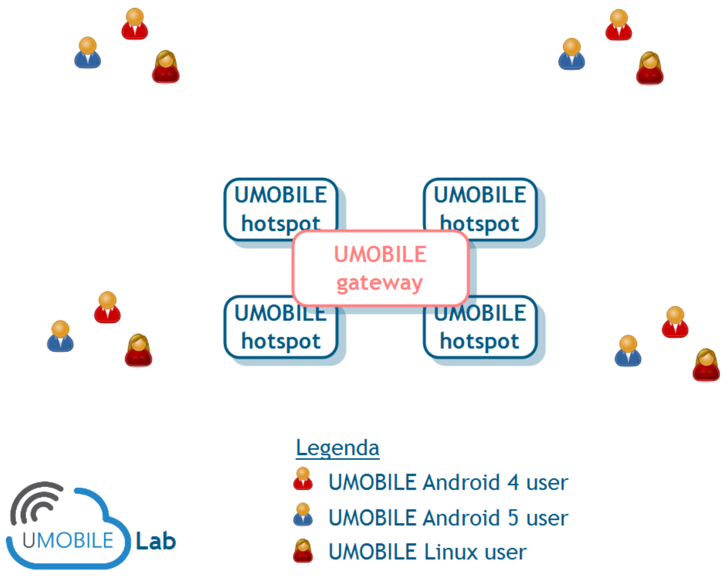


Figure 11. UMOBILE testbed: simplified architecture

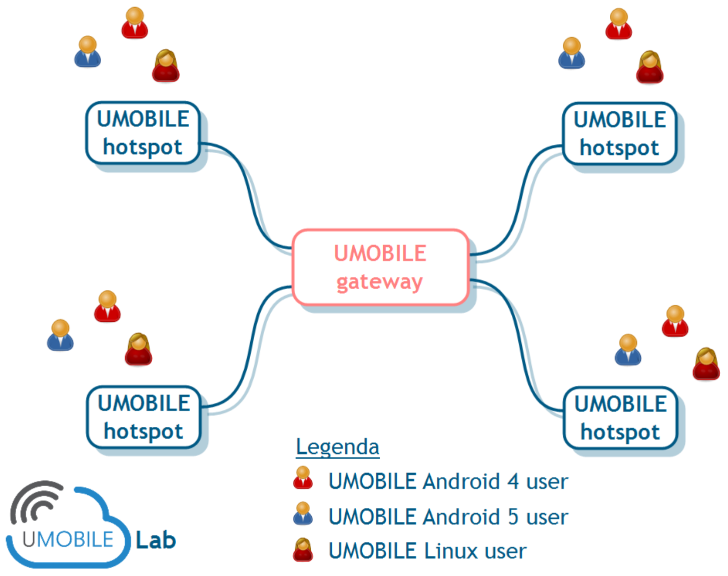


Figure 12. UMOBILE testbed: more realistic architecture

The detailed complete architecture is represented in Figure 13.

In the testbeds, the human users and their devices are replaced by **Android and Linux black-boxes**; these black-boxes will receive commands from a **test robot**, i.e., a software program issuing defined test sequences of commands to the applications running on the black-boxes. The test robot is depicted on the right side of Figure 13. The test robot is connected to each black-box through the **test network**.

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 645124

Each element in the lab is here described in its hardware and software components. The elements and their components may be modified following the progress of the project.
For specific trials, some sections of the lab may also be kept down. Implementation issues will be considered, such as the number of usable channels in a single area (max 3) in the 2.4 GHz range (802.11n - and 'only n' - should overcome this issue).
Two kinds of Android black boxes will be used, in order to support two releases of the Android OS (e.g. 4.2 and 5.1). It will be possible to exchange data directly through Wi-Fi Direct, re-purposing some of the Android devices.
Both APRs (Access Points) and CAPRs (Core Access Points) devices have 2.4 Wi-Fi radios, act as access points and are complemented with SEG (Service Execution Gateway - Raspberry PI). The black boxes (i.e., the UMOBILE users) can connect and move among all CAPRs and APRs.

The users will be a part of the lab with their handheld devices. A massive users' activity will be simulated during the trials, issuing automatic sequences of commands to:

- Android-based black boxes, based on BananaPI Android devices, that have been selected for their capability to run apps based on the IBR-DTN library version 1.01. Namely the devices are:
    - A4BB: Android 4.2 on BPI-M2 BananaPI device, based on ARM A31S, with 1 x GEthernet and 1 x Wi-Fi 802.11 b/g/n
    - A5BB: Android 5.1 on BPI-M3 BananaPI device, based on ARM A83T, with 1 x GEthernet and 1 x Wi-Fi 802.11 b/g/n
    - Linux based black-boxes (LBB): Raspberry PI 3 with a Raspbian distro from https://www.raspberrypi.org/downloads/raspbian/ (RASPBIAN JESSIE 2016-05-10 Kernel version: 4.4).

A mini-server (the *test robot*) will issue the sequences of commands constituting the trial.
The separate test network will reach every user device; the test network will use separate links (over different wires or different VLAN), in order not to interfere with the dataflows generated and observed during the trials.

In the right upper corner of the image, the **VPN concentrator** is depicted, with its connections to the devices of the testbed, through the **management network**; these connections will allow the members of the project to access, control and modify each device, possibly flashing it with new versions of *UMOBILEized* software.

The management network allows project members to reach any device in the testbed, through an IPV4 VPN over the Internet (L2TP or OpenVPN).

Please refer to Annex A for details of Lab implementation and configuration, and for usage instructions.

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 645124

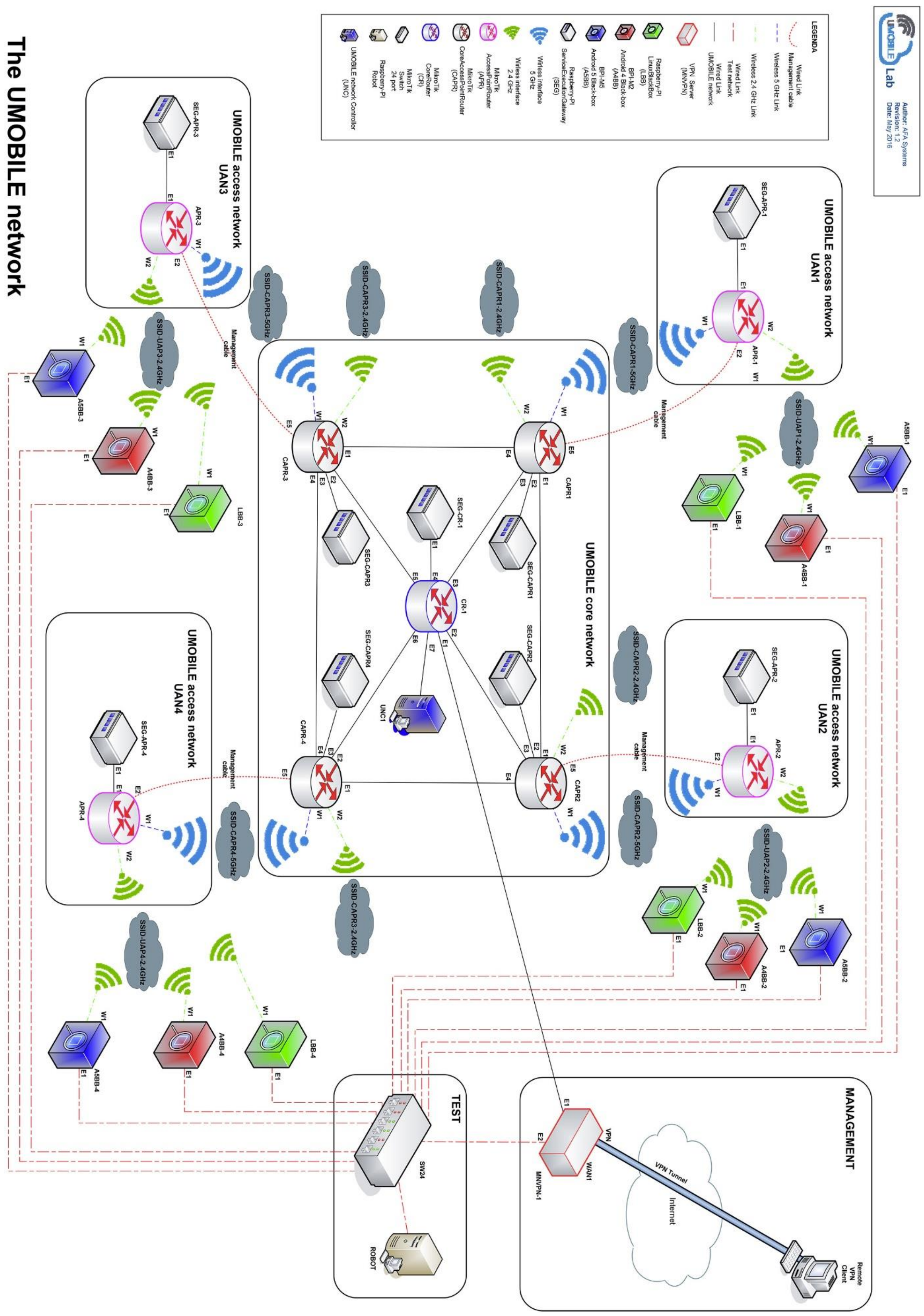


Figure 13. UMOBILE Lab - Detailed architecture

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 645124

Figure 14 shows the physical installation of the Lab in Termoli (Italy).

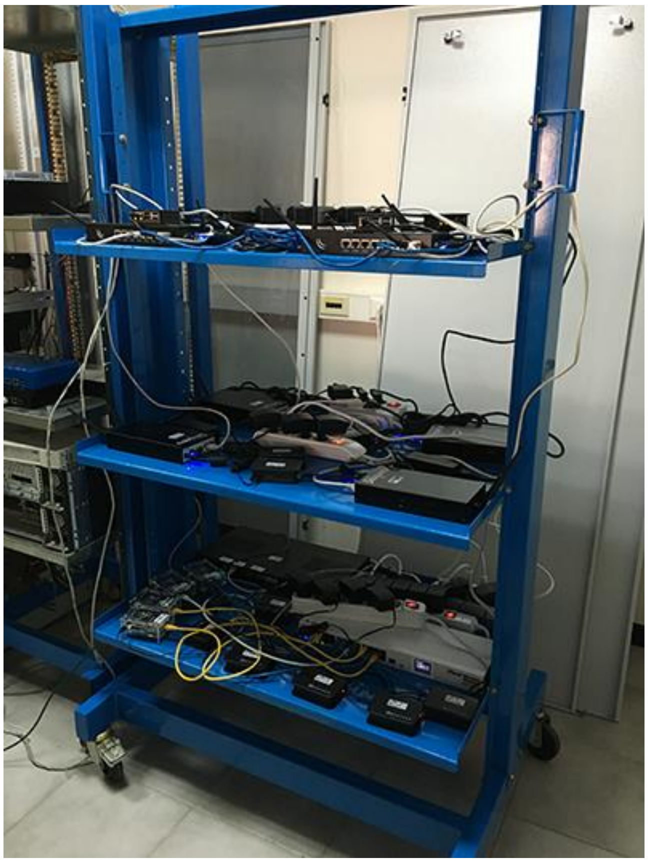


Figure 14. UMOBILE testbed - physical installation

3.2. Devices

Table 1 shows the lab components, specifying for each device the acronym used in the detailed architecture image (Figure 13). The technical specification of each device is reported in Table 2.

Device	Usage	Ethernet	Additional ports	USB port	role in UM Lab	UM acronym
RB493G	infrastructure	9 GE	serial	y	CORE NETWORK	CRCAPR
RB435G	infrastructure	3 GE	serial	y	ACCESS NETWORK	APR
R52n-M	infrastructure	-	-	-	radio add-on	-
Raspberry Pi3	infrastructure	1	HDMI	y	SERVICE EXECUTION	SEG
Raspberry Pi3	hand-held device emulation	1	HDMI	y	USER DEVICE	LBB
BPI-M2	hand-held device emulation	1	HDMI	y	USER DEVICE	A4BB
BPI-M3	hand-held device emulation	1	HDMI	y	USER DEVICE	A5BB

Table 1. Lab components overview

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 645124

Type	Name	Model	Accessories	q.ty	Description
CoreRouter (CR)	CR				CoreRouter (CR), 9 Giga Ethernet
		RB493G		1	Nine Gigabit ethernet ports, three miniPCI slots http://routerboard.com/RB493G
			CA493	1	Black aluminium Indoor case (3 holes for Nfemale Bulkhead connectors or AC/SWI Swivel antennas) http://routerboard.com/CA493
			MicroSD	1	MicroSD 32G
			24HPOW	1	24V 1.6A adapter, for long cables, for installations with many miniPCI http://routerboard.com/24HPOW
CoreAccessPointRouter (CAPR)	CAPR				CoreAccessPointRouter (CAPR) 9 Giga Ethernet, 1 x wireless dual-channel 5 Ghz, 1 x wireless dual-channel 2.4 Ghz
		RB493G		1	Nine Gigabit ethernet ports, three miniPCI slots http://routerboard.com/RB493G
			CA493	1	Black aluminium Indoor case (3 holes for Nfemale Bulkhead connectors or AC/SWI Swivel antennas) http://routerboard.com/CA493
			MicroSD	1	MicroSD 32G
			RBGPOE	1	The new MikroTik Gigabit PoE adapter for powering any RouterBOARD over Gigabit Ethernet http://routerboard.com/RBGPOE
			24HPOW	1	24V 1.6A adapter, for long cables, for installations with many miniPCI http://routerboard.com/24HPOW
			R52nM	2	802.11a/b/g/n dual band miniPCI card Two MMCX antenna connectors http://routerboard.com/R52nM
			ACSWIM	4	2.4-5.8GHz Swivel Antenna with cable and MMCX connector. http://routerboard.com/ACSWIM
AccessPointRouter (APR)	APR				AccessPointRouter (APR) 3 Giga Ethernet, 1 x wireless dual-channel 5 Ghz, 1 x wireless dual-channel 2.4 Ghz
		RB435G		1	Five miniPCI slots and three Gigabit Ethernet ports http://routerboard.com/RB435G
			CA433U	1	Indoor case (3 holes for Nfemale Bulkhead connectors or Swivel antennas and hole for USB http://routerboard.com/CA433U
			MicroSD	1	MicroSD 32G
			RBGPOE	1	The new MikroTik Gigabit PoE adapter for powering any RouterBOARD over Gigabit Ethernet http://routerboard.com/RBGPOE
			24HPOW	1	24V 1.6A adapter, for long cables, for installations with many miniPCI http://routerboard.com/24HPOW
			R52nM	2	802.11a/b/g/n dual band miniPCI card Two MMCX antenna connectors http://routerboard.com/R52nM
			ACSWIM	4	2.4-5.8GHz Swivel Antenna with cable and MMCX connector. http://routerboard.com/ACSWIM
ServiceExecutionGateway (SEG)	SEG				ServiceExecutionGateway (SEG) RASPBERRY PI 3 MODEL B 1 Ethernet, 1 x wireless dual-channel 2.4 Ghz
		RASPBERRY PI 3		1	RASPBERRY PI 3 MODEL B https://www.raspberrypi.org/products/raspberry-pi-3-model-b/
			RasbianDoker		http://blog.hypriot.com/downloads/

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 645124

			power supply	1	Micro USB Plug In Power Supply http://uk.rs-online.com/web/p/plug-in-power-supply/9098135/
			Case Black	1	Raspberry Pi 3 B Development Board Case, Black, Grey http://uk.rs-online.com/web/p/development-board-enclosures/9098138/
			MicroSD	1	MicroSD 32G
UmobileNetworkController (UNC)	UNC				UmobileNetworkController (UNC) Server + Storage
Android 5 Black-box (A5BB)	A5BB				AndroidBlackBox 5.x (A5BB) BPI-M3 1 x ethernet , 1 x wireless dual-channel 2.4 Ghz
		BPI-M3		1	5.1 Android Version http://www.bananapi.com/index.php/component/content/article?layout=edit&id=85
			Android 5.1		http://www.bananapi.com/index.php/download?layout=edit&id=90
			MicroSD	1	MicroSD 32G
			Case	1	Trasparent Case
			power supply	1	power supply
			ACSWI	1	2.4/5Ghz Omni antenna, 4dBi gain, u.fl connector http://routerboard.com/ACSWI
Android 4 Black-box (A4BB)	A4BB				AndroidBlackBox 4.x (A4BB) BPI-M2 1 x ethernet , 1 x wireless dual-channel 2.4 Ghz
		BPI-M2		1	4.2 Android Version http://www.bananapi.com/index.php/component/content/article?layout=edit&id=73
			Android 4.2		http://www.bananapi.com/index.php/download?layout=edit&id=75
			MicroSD	1	MicroSD 32G
			Case	1	Trasparent Case
			power supply	1	power supply
			ACSWI	1	2.4/5Ghz Omni antenna, 4dBi gain, u.fl connector http://routerboard.com/ACSWI
Linux Black-box (LBB)	LBB				LinuxBlackBox (LBB) RASPBERRY PI 3 MODEL B 1 Ethernet, 1 x wireless dual-channel 2.4 Ghz
		RASPBERRY PI 3		1	RASPBERRY PI 3 MODEL B https://www.raspberrypi.org/products/raspberry-pi-3-model-b/
			Rasbian		Raspbian distro from https://www.raspberrypi.org/downloads/raspbian/(RASPBIAN JESSIE 2016-05-10 Kernel version:4.4)
			power supply	1	Micro USB Plug In Power Supply http://uk.rs-online.com/web/p/plug-in-power-supply/9098135/
			Case Black	1	Raspberry Pi 3 B Development Board Case, Black, Grey http://uk.rs-online.com/web/p/development-board-enclosures/9098138/
			MicroSD	1	MicroSD 32G
Server Test Robot (ROBOT)	ROBOT	Same of LBB			Server Test Robot (ROBOT) RASPBERRY PI 3 MODEL B 1 Ethernet, 1 x wireless dual-channel 2.4 Ghz
Switch 24 Porte	SW24	CRS125		1	Switch Mikrotik 24

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 645124

		-24G-1S-RM			ports http://routerboard.com/CRS125-24G-1S-RM
MajorNET VPN Server (MNVPN)	MNVPN	MajorNET	MNP364	1	MajorNET VPN Server (MNVPN) http://www.majornet.it/funzioni/majorgw

Table 2. Lab components specifications

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 645124

4. Proof of Concept

4.1. Overview

In order to demonstrate the performance and efficiency of UMOBILE architecture within a testbed, we have selected some specific use-cases identified in the WP2 for the purpose of feature demonstration. More specifically, we focus on two demonstrations: the first one covers both the emergency and the civil protection scenarios, and the second demonstration targets the service announcement and social routine scenarios.

4.1.1. Schema

In order to get a better understanding of the two POCs defined in UMOBILE, we use a fixed schema to describe the different aspects to take into account. Figure 15 depicts a mind map with all the concepts involved in the POC sections. A brief explanation follows.

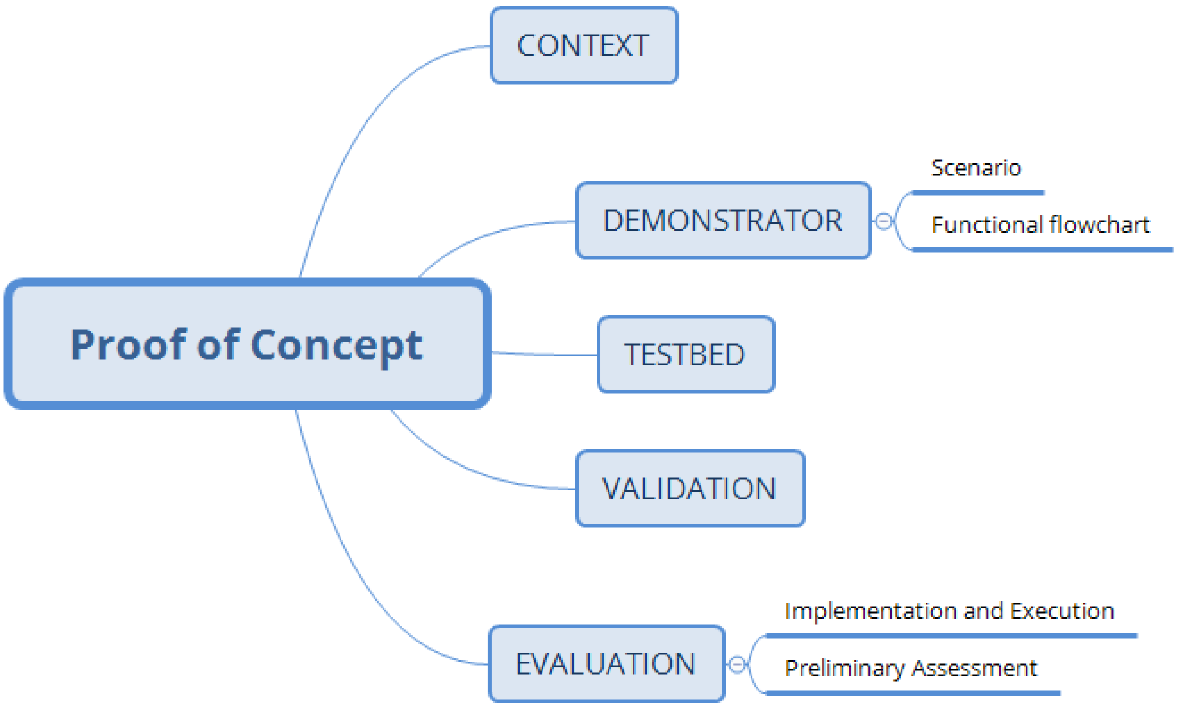


Figure 15. Proof of Concept schema

CONTEXT: The context section provides a general overview of the proof of concept.

DEMONSTRATOR: The demonstrator is a description of the identified use case. The **scenario** describes the story of the demo; the definition of the **functional flowchart** addressed in the POC is also provided.

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 645124

TESTBED: This section describes the lab configuration necessary to demonstrate the POC.

VALIDATION PLAN: The validation plan is a list of actions and activities to perform on the testbed in order to validate the POC.

EVALUATION: The evaluation refers to the technical **implementation** and deployment of the POC and the **results and conclusions** obtained with it. The evaluation phase will be in charge of verifying the identified requirements. For the time being, the evaluation phase is not yet started, and it will be fully described in D.5.4.

4.2. POC 1: Emergency and Civil scenario

4.2.1. Context

In the first proof of concept, we present the services introduced by UMOBILE for reaching disconnected areas and users. UMOBILE provides mechanisms that may assist responsible authorities in the case of challenged events. In emergencies, traditional communication services such as fixed or mobile networks and local Internet access are completely/partially inoperable. Therefore, the design of the UMOBILE system assists users in disseminating emergency information directly via end-user devices as well as via the UMOBILE hotspots and UAVs.

This proof of concept is focused on displaying the capability of the UMOBILE architecture to reach disconnected areas and assist responsible authorities in challenged events, corresponding to an infrastructure scenario.

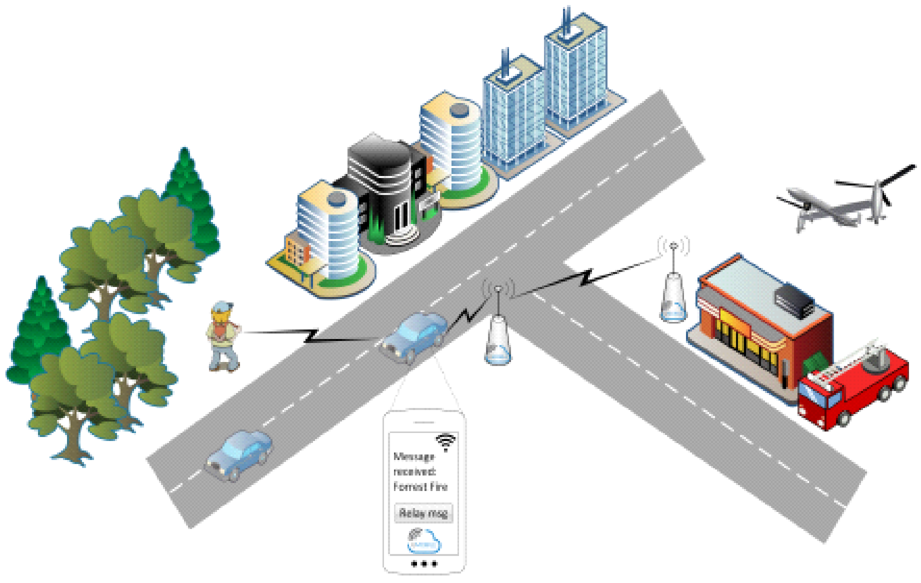


Figure 16. Emergency scenario & emergency message dissemination

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 645124

4.2.2. Demonstrator

4.2.2.1. Scenario

We consider the following scenario, illustrated in Figure 17, in which a user is enjoying a day in a national forest. There is limited internet access as this network topology is sparse showing participations in time and space. This person at a given point in time spots a fire and wants to alert about it. The user tags this info as “emergency” and use UMOBILE to make it available to be shared globally, as he has no cellular coverage. After that, the user continues his walk in an alternative path close to main roads, where bikers are passing by. The message is stored for a given amount of time in a specific geographic location. Bikers, by means of their UMOBILE app, will collect any emergency info by default. As soon as any of the UMOBILE bikers have connection, they will pass the message towards an authorized emergency entity via UMOBILE system, in an opportunistic fashion.

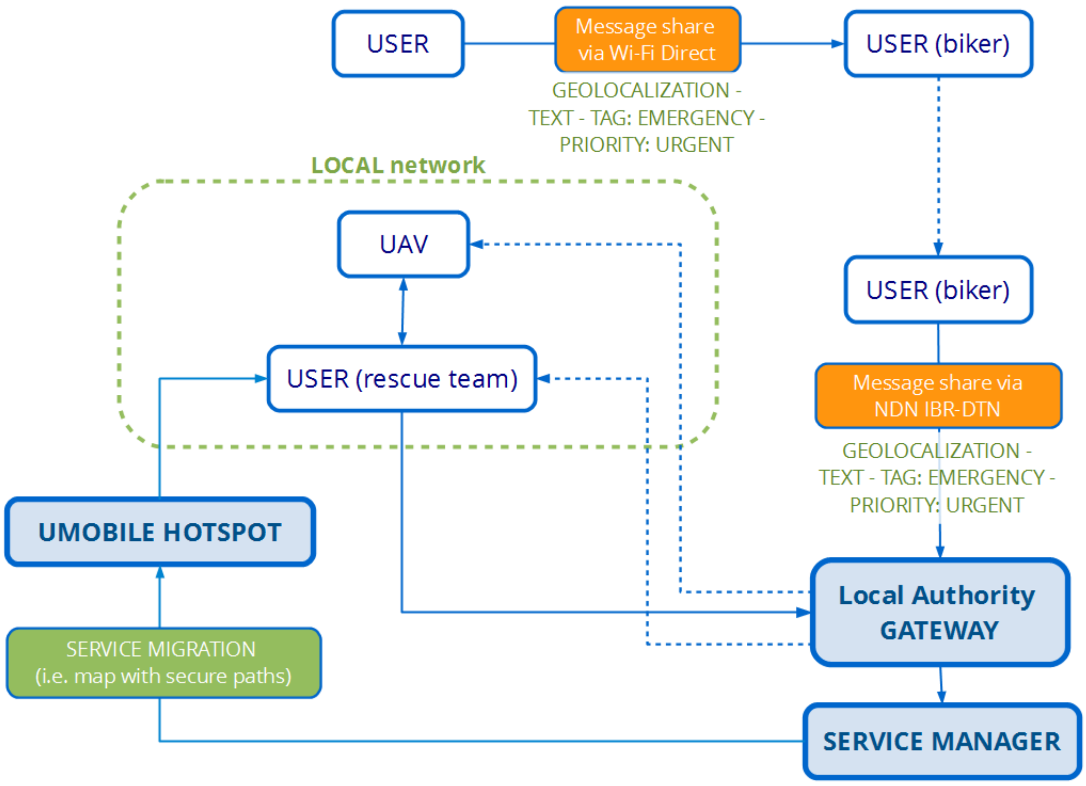

USER
Message share via Wi-Fi Direct
USER (biker)
GEOLOCALIZATION - TEXT - TAG: EMERGENCY - PRIORITY: URGENT
LOCAL network
UAV
USER (biker)
USER (rescue team)
Message share via NDN IBR-DTN
GEOLOCALIZATION - TEXT - TAG: EMERGENCY - PRIORITY: URGENT
UMOBILE HOTSPOT
Local Authority GATEWAY
SERVICE MIGRATION (i.e. map with secure paths)
SERVICE MANAGER

Figure 17. High-level flowchart of the first Proof of concept

Upon reception of such message, a team of fire fighters is sent to the location, equipped with UMOBILE apps for communication purposes. They use a UAV equipped with Wi-Fi to create a local communication infrastructure: the UAV is instructed to keep a formation able to cover the complete area affected by the fire, i.e., working as an access point for the rescue team. The video camera on the UAV provides with real-time images of the ground to the control units in the field.

Members of the rescue teams, furthermore, use their mobile devices to request the

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 645124

authority (which acts as a Content provider) to deploy an emergency service in the affected area, such as a map service with path recommendations to the user. The request eventually reaches the authority that is willing to do its best to deploy the requested service.

The authority is assumed to be in possession of information about the current network's condition (links availability, traffic, congestion, etc.) and potential number of users. Likewise, it is assumed to be in possession of CPU, disk and memory resources of his UMOBILE Hotspots. For example, he has accurate information about the total, free and used memory of each Hotspot. This information is available to the Service Manager, which is equipped with a Decision Engine.

The Decision Engine is equipped with algorithms that, based on the issued requests, the status of the network and of the Hotspots, can determine the best Hotspots to deploy the service.

The selected Hotspots (one or more) instantiate the service that will proceed to respond to requests issued by members of the rescue teams and the victims.

4.2.2.2. Functional flowchart

From a functional point of view, the defined scenario can be mapped into the flowchart described in Figure 18.

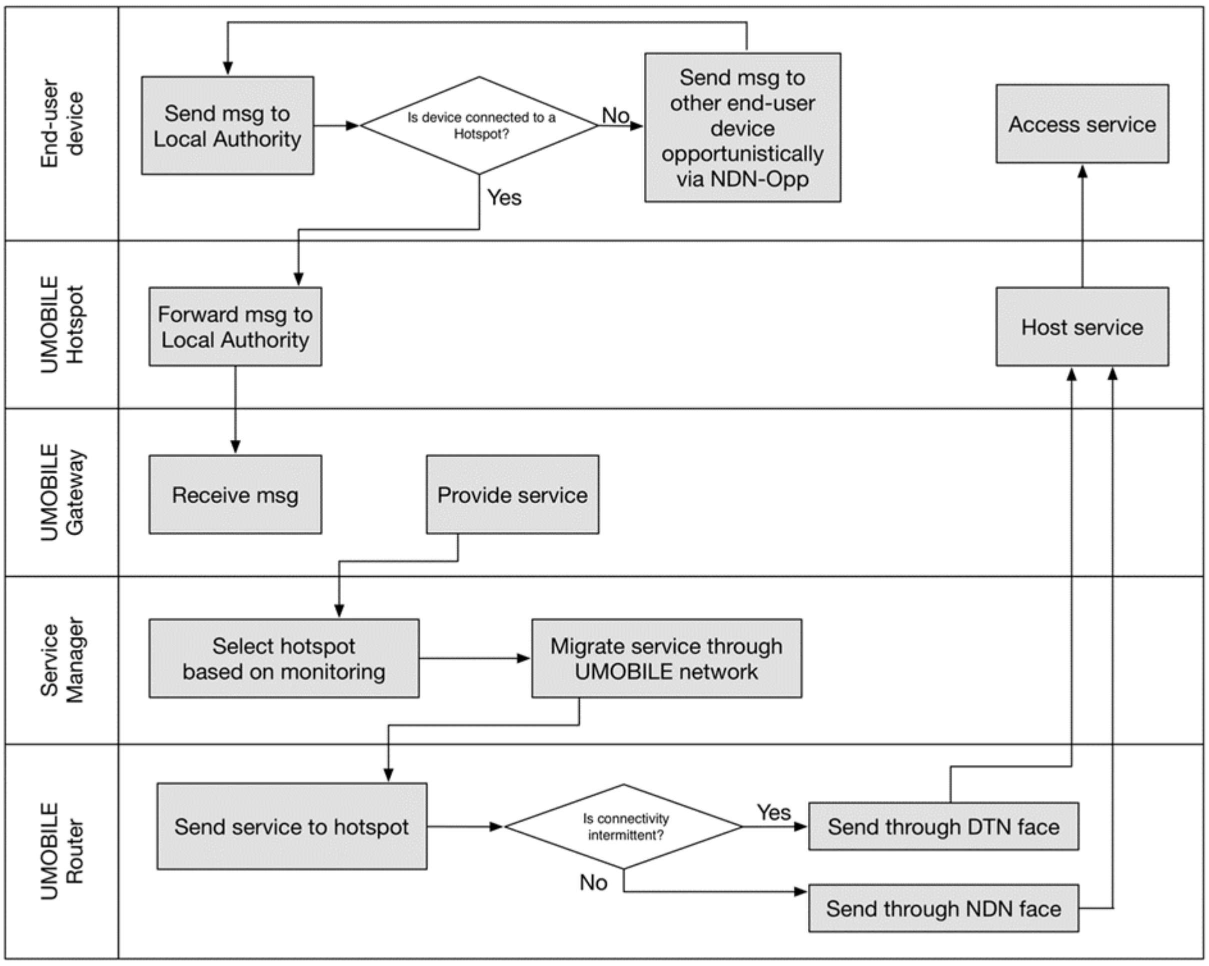

End-user device
Send msg to Local Authority
Is device connected to a Hotspot?
No
Yes
Send msg to other end-user device opportunistically via NDN-Opp
Access service
UMOBILE Hotspot
Forward msg to Local Authority
Host service
UMOBILE Gateway
Receive msg
Provide service
Service Manager
Select hotspot based on monitoring
Migrate service through UMOBILE network
UMOBILE Router
Send service to hotspot
Is connectivity intermittent?
Yes
No
Send through DTN face
Send through NDN face

Figure 18. Functional flowchart POC1

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 645124

A list of requirements which need to be satisfied by the proof of concept, based on D.2.2, can be defined:

- UMOBILE APP SHOULD be compatible with a specific set of Device Feature to be defined
- UMOBILE APP SHOULD be compatible with a specific API level to be defined
- UMOBILE APP MUST be able to SEND and TAG (for example as “emergency”) A MESSAGE to UMOBILE SYSTEM
- UMOBILE APP MUST be able to INFER USER CONTEXT (geo-location) to complement the EMERGENCY MESSAGGE.
- UMOBILE APP SHOULD be able to exploit every COMMUNICATION OPPORTUNITY to send the message
- UMOBILE APP MUST be able to SHARE the EMERGENCY MESSAGE among UMOBILE USERS using Wi-Fi Direct
- UMOBILE APP MUST be able to SHARE MESSAGES among UMOBILE USERS based on users’ PREFERENCES and interaction in the system (i.e. subscription to “emergency services”), ensuring user privacy
- UMOBILE APP MUST be able to STORE the EMERGENCY MESSAGE until a UMOBILE USER is available to receive the message
- UMOBILE APP MUST be able to SHARE the EMERGENCY MESSAGE over NDN and IBR-DTN
- UMOBILE GATEWAY SHOULD be able to PERFORM DATA FUSION, combining different pieces of data aiming to generate more reliable emergency information
- UMOBILE SYSTEMS over UAV MUST be able to provide a local communication Wi-Fi infrastructure
- UMOBILE APP MUST be able to differentiate between messages shared by UMOBILE users and messages shared by authorities
- UMOBILE HOTSPOTS MUST be able to COLLECT, STORE and RELAY relevant information (e.g., alert messages, instructions from emergency authorities)
- UMOBILE HOTSPOTS SHOULD be able to PERFORM DATA FUSION, combining different pieces of data aiming to generate more reliable emergency information
- UMOBILE SERVICE MANAGER MUST be able to migrate some services from the Internet and operate locally within the UMOBILE domain.

4.2.3. Testbed

Figure 19 shows the UMOBILE Lab configuration set up in order to validate the first proof of concept.

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 645124

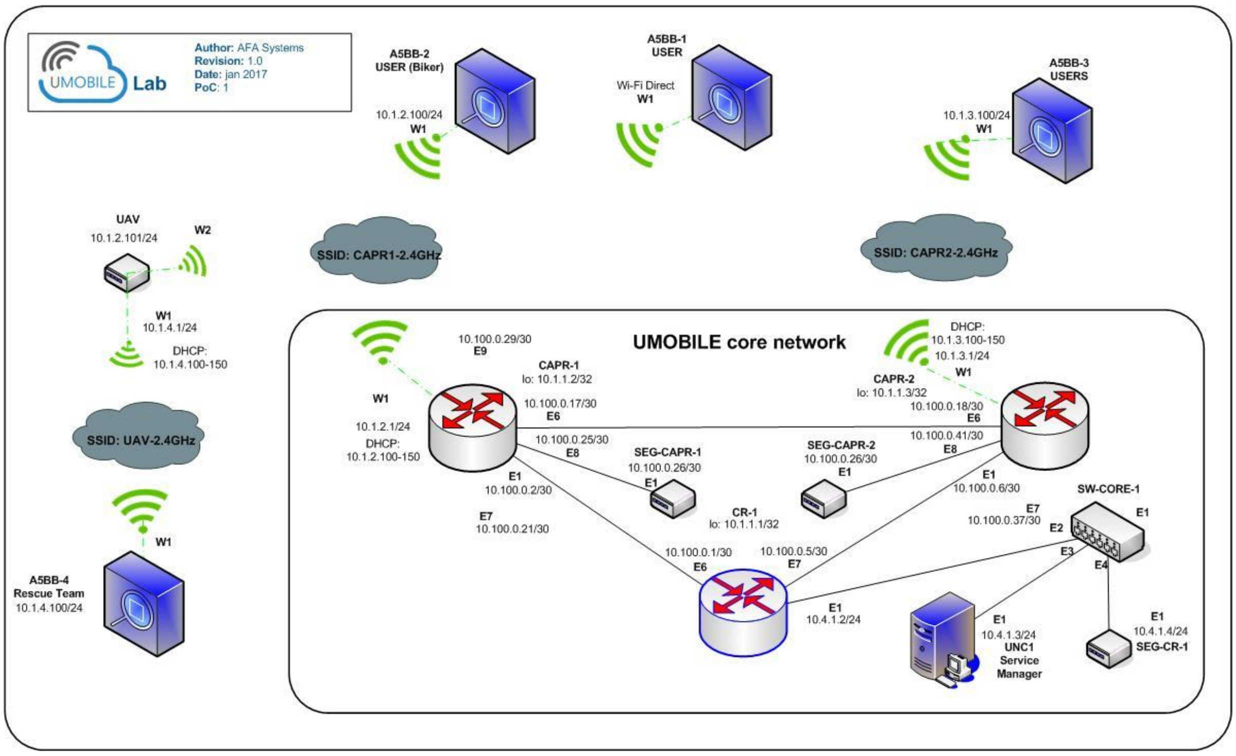


Figure 19. UMOBILE Lab configuration POC1

4.2.4. Validation

1. Compose an emergency message (write text, add tag, automatically add geo-localization) with UMOBILE app from android black-box A5BB-1
2. Keep down all the others devices (no internet coverage)
3. Send the emergency message via Wi-Fi Direct (the message is stored until a receiver is available)
4. Enable the android black-box A5BB-2, in order to receive the emergency message
5. Keep down android black-box A5BB-1
6. Enable the gateway CR-1
7. The gateway receives the emergency message via NDN IBR-DTN
8. Enable Linux-box UAV and setup local access point through connection to gateway
9. Compose a message with UMOBILE app from android black-box A5BB-4 and send it to UAV via Wi-Fi W2
10. The message is sent to gateway via Wi-Fi W1
11. Enable all the other devices
12. The service manager E1 checks the available hotspots
13. The service manager migrates a specific service to the selected hotspot CAPR-2
14. Compose a message with UMOBILE app from the gateway CR-1 and send it to UMOBILE hotspot CAPR-2
15. The hotspot CAPR-2 forwards the message to user A5BB-3

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 645124

4.3. PoC 2: Service Announcement and Social-Routine scenario

4.3.1. Context

In this proof of concept, we present how the UMOBILE architecture can provide new types of services, such as micro-blogging and social-routine, showcasing the benefits of opportunistic communications and enhancements to smart routing and QoS aware services.

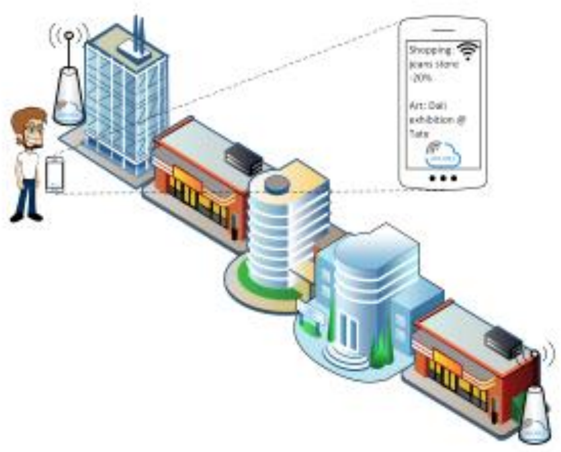


Figure 20. Service Announcement and Social-Routine scenario

4.3.2. Demonstrator

4.3.2.1. Scenario

In this demonstrator, a UMOBILE user, as illustrated in Figure 21, generates and shares an expression of interest in the form of tagged information (e.g., nearby supermarket offers or restaurant review), by means of the UMOBILE app. The message is sent to the UMOBILE system where all UMOBILE users can benefit from the local information. In addition, the UMOBILE system will keep track of the local event and update useful information to the UMOBILE subscribed users based on the new created information and its relevant hashtag. This information can be referred to the availability of some form of service (QoS aware service), as well as to user experience and user perception shared among users.

UMOBILE users will be able to exchange and share information, e.g., photos, videos, social text comments among the contacts in their trust circles derived from Wi-Fi Direct as well as Bluetooth. User devices do not have to be always connected to the Internet. Data is exchanged among people passing by, based on social interaction approaches. The UMOBILE system takes advantage of all available connectivity (e.g., wired or wireless) and intelligently chooses the most suitable transmission protocol depending on the network environment.

UMOBILE hotspots, deployed along this scenario, will store data based on its local

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 645124

meaningfulness, e.g. tips/photos about nearby shops, and disseminate among other peering hotspots or UMOBILE users.

Furthermore, the UMOBILE system will provide the user with additional information regarding their personal expressed interests and, e.g., estimation of the time to get to a certain suggested location based on history information.

Another crucial aspect supported by UMOBILE is the capability to capture personal data of UMOBILE users (e.g. visited networks, affinity network) with the ultimate goal of improving the user's routine.

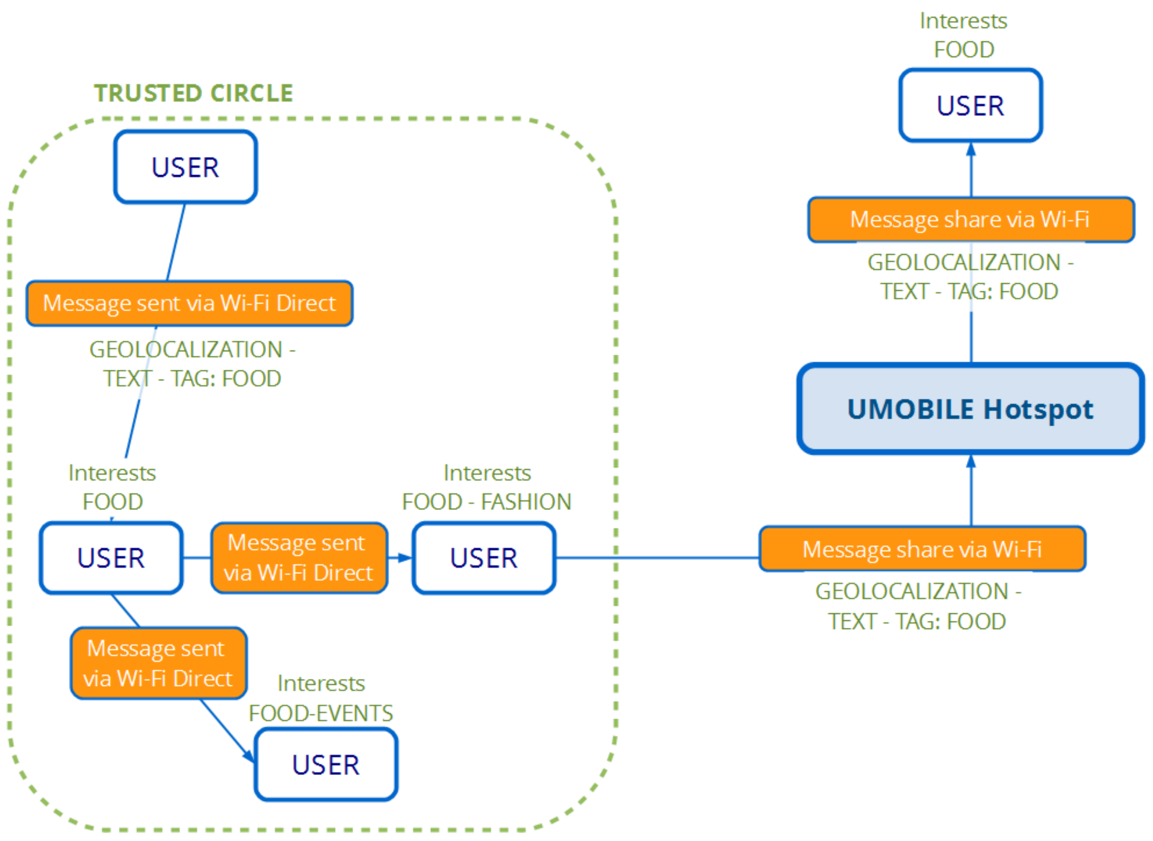

TRUSTED CIRCLE
USER
Message sent via Wi-Fi Direct
GEOLOCALIZATION - TEXT - TAG: FOOD
Interests FOOD
USER
Message sent via Wi-Fi Direct
Interests FOOD - FASHION
USER
Message sent via Wi-Fi Direct
Interests FOOD-EVENTS
USER
Interests FOOD
USER
Message share via Wi-Fi
GEOLOCALIZATION - TEXT - TAG: FOOD
UMOBILE Hotspot
Message share via Wi-Fi
GEOLOCALIZATION - TEXT - TAG: FOOD

Figure 21. High-level flowchart of the second Proof of concept

4.3.2.2. Functional flowchart

From a functional point of view, the defined scenario can be mapped into the flowchart described in Figure 22.

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 645124

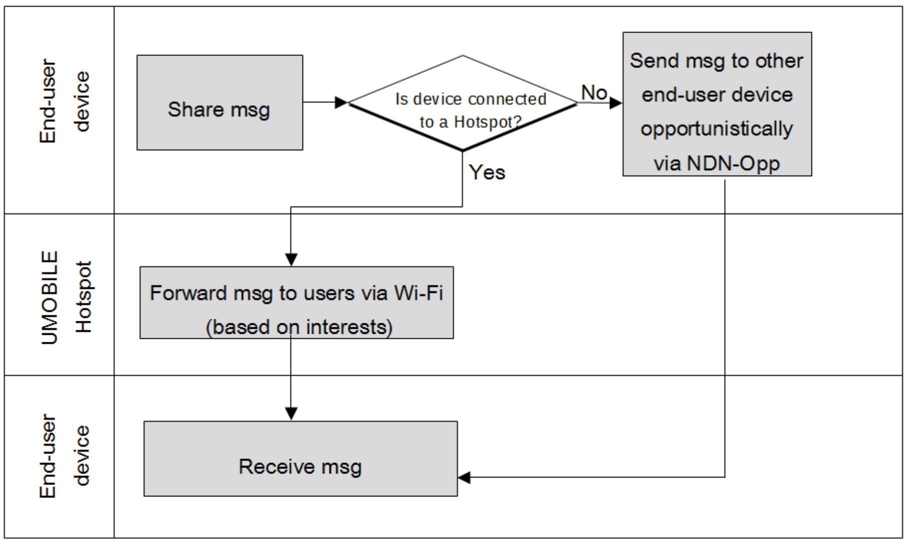

End-user device
UMOBILE Hotspot
End-user device
Share msg
Is device connected to a Hotspot?
No
Yes
Send msg to other end-user device opportunistically via NDN-Opp
Forward msg to users via Wi-Fi (based on interests)
Receive msg

Figure 22. Functional flowchart POC2

A list of requirements, which need to be satisfied by the proof of concept, based on D.2.2, can be defined:

- UMOBILE APP SHOULD be compatible with a specific set of Device Feature to be defined
- UMOBILE APP SHOULD be compatible with API level to be defined
- UMOBILE APP MUST be able to SEND and TAG (for example as "events") A MESSAGE to UMOBILE SYSTEM
- UMOBILE APP MUST be able to INFER USER CONTEXT (geo-location) to complement the MESSAGGE.
- UMOBILE APP SHOULD be able to exploit every COMMUNICATION OPPORTUNITY to send the message
- UMOBILE APP MUST be able to SHARE the MESSAGE among UMOBILE USERS using Wi-Fi Direct
- UMOBILE APP MUST be able to SHARE the MESSAGE among UMOBILE USERS using Bluetooth
- UMOBILE APP MUST be able to SHARE MESSAGES among UMOBILE USERS based on users' PREFERENCES and interaction in the system (i.e. subscription to "food"), ensuring user privacy.
- UMOBILE APP MUST be able to STORE the MESSAGE until a UMOBILE USERS is available to receive the message
- UMOBILE HOTSPOTS MUST be able to COLLECT, STORE and RELAY relevant information, based on user circles and preferences
- UMOBILE HOTSPOTS SHOULD be able to PERFORM DATA FUSION, combining different pieces of data aiming to generate more reliable emergency information

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 645124

- UMOBILE GATEWAY SHOULD be able to PERFORM DATA FUSION, combining different pieces of data aiming to generate more reliable emergency information

4.3.3. Testbed

Figure 23 shows the UMOBILE Lab configuration set up in order to validate the second proof of concept.

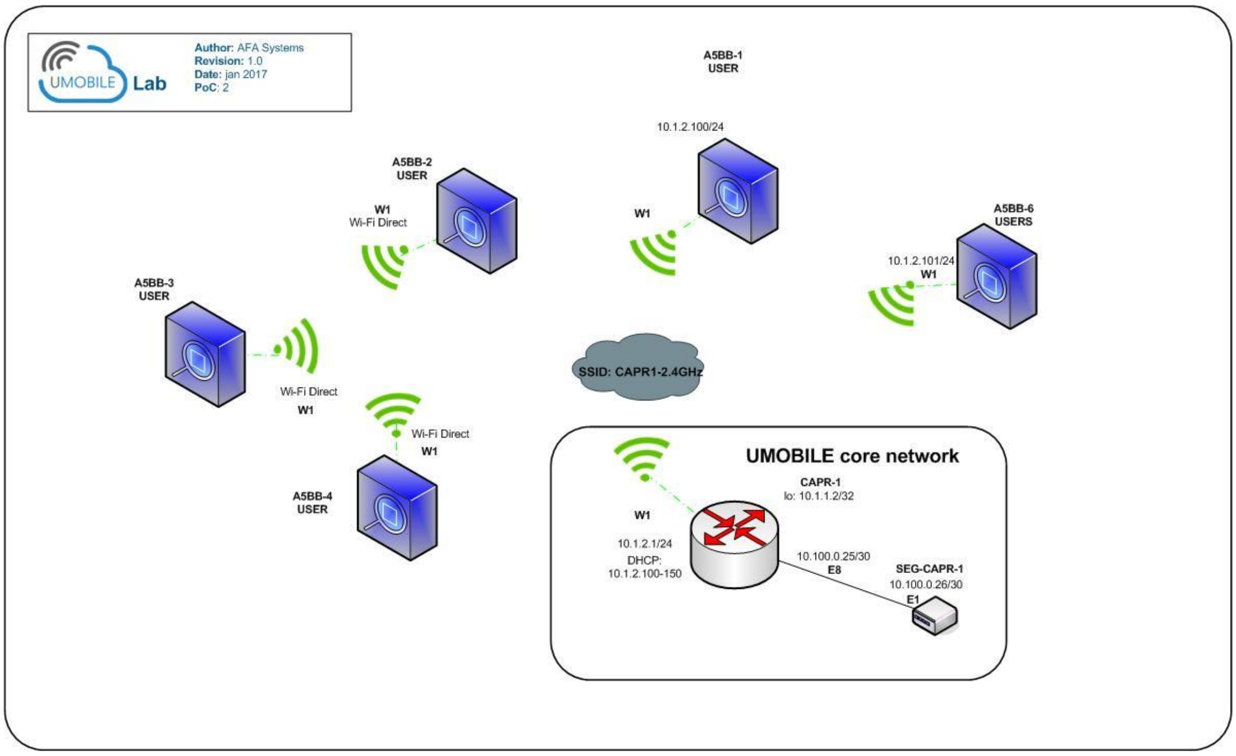


Figure 23. UMOBILE Lab configuration POC2

4.3.4. Validation

1. Compose a message with UMOBILE app from android black box A5BB-2: write text, add tag/keywords (i.e.: "food"), automatically add geo-localization.
2. Send the message via Wi-Fi Direct to users (A5BB-3 and A5BB-4) which satisfy the following requirements: they are IN THE TRUSTED CIRCLE; their INTERESTS are relevant to the message (i.e.: based on the tag)
3. Send again the message using Bluetooth
4. Share the message with local UMOBILE enabled hotspot CAPR-1 using 3G connectivity
5. Share the message through Wi-Fi to user A5BB-6 which satisfy the following requirements: they are IN THE TRUSTED CIRCLE; their INTERESTS are relevant to the message (i.e.: based on the tag)

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 645124

5. Conclusions

In this deliverable, we have provided a functional description of the **integration of UMOBILE system**. The UMOBILE devices are defined; the interaction between software modules is described and will be then evaluated through the proof-of-concept software.

The **proof-of-concept** is created based on two specific demos: two independent technological demonstrations based on software developed during UMOBILE, and which rely on two of the use-cases selected in WP2. The proof-of-concept is initially described in this deliverable; then, in M34, the full aspects concerning the proof-of-concept and its evaluation will be described and reported in D5.4 “Proof-of-Concept (2)”.

Furthermore, the present deliverable includes also a short guideline to the usage of the UMOBILE Lab, which will be used as testbed for the proof-of-concept, as well as details on the implementation of the Lab.

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 645124

6. References

[1] UMOBILE Project, “D.3.3 - UMOBILE ICN layer abstraction initial specification”, February 2016

[2] UMOBILE Project, “D.3.1 - UMOBILE architecture report (1)”, May 2016

[3] UMOBILE Project, “D.2.2 - System and Network Requirements Specification”, March 2016

[4] L. Amaral, R. C. Soa, P. Mendes, and W. Moreira, “Oi! – opportunistic data transmission based on Wi-Fi direct,” in IEEE Infocom 2016 Live/Video Demonstration (Infocom’16 Demo), (San Francisco, USA), Apr. 2016.

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 645124

Annex A. UMOBILE Lab: Usage and Implementation

A1. Usage

The UMOBILE Lab is composed by two networks, to permit different kinds of tests. We have:

- UMOBILE network, IP routed by the OSPF protocol;
- TEST network, for devices handling (over ethernet), in order to guarantee accessibility during the change of the UMOBILE Lab wireless network.

To access the UMOBILE Lab it is necessary an L2TP/IPSec connection with the following data:

```
Server: lab1.umobile-project.eu
VPN Type: L2TP/IPSec (Layer 2 Tunneling Protocol with IPsec)
Advanced properties use preshared Key: password
Authentication Protocol allow only: PAP
```

A1.1 Device Access

Once logged into UMOBILE Lab, it is possible to access:

- **each device in the lab**; their IP addresses and hostname are listed in UMOBILE Lab devices table (see Table 3);
    - the "UMOBILE Lab devices" table contains the links to the devices which has a web GUI;
    - each device can access the Internet;
    - each device can be access via SSH;
- **UMOBILE website** and **UMOBILE project-wiki**;
- the **Internet**, through a captive portal (see Figure 24), using the same username/password as L2TP;
- the **Drive**, a common document repository.

Nr.	IP	Hostname	OS	zone	ssh access	http access
1	10.1.1.2	CAPR-1	OpenWRT CHAOS CALMER (15.05.1, r48532) – AP	core	ssh://capr-1.umobilelab.local /	http://capr-1.umobilelab.local
2	10.1.1.3	CAPR-2	OpenWRT CHAOS CALMER (15.05.1, r48532) – AP	core	ssh://capr-2.umobilelab.local	http://capr-2.umobilelab.local
3	10.1.1.4	CAPR-3	OpenWRT CHAOS CALMER (15.05.1, r48532) – AP	core	ssh://capr-3.umobilelab.local	http://capr-3.umobilelab.local
4	10.1.1.5	CAPR-4	OpenWRT CHAOS CALMER	core	ssh://capr-4.umobilelab.local	http://capr-4.umobilelab.local

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 645124

			(15.05.1, r48532) – AP			
5	10.4.1.2	CR-1	OpenWRT CHAOS CALMER (15.05.1, r48532) – AP	core	ssh://cr-1.umobilelab.local	http://cr-1.umobilelab.local
6	10.100.0.26	SEG-CAPR-1	Raspbian 8 (jessie) HypriotOS – Docker – AP	core	ssh://seg-capr-1.umobilelab.local	
7	10.100.0.42	SEG-CAPR-2	Raspbian 8 (jessie) HypriotOS – Docker – AP	core	ssh://seg-capr-2.umobilelab.local	
8	10.100.0.58	SEG-CAPR-3	Raspbian 8 (jessie) HypriotOS – Docker – AP	core	ssh://seg-capr-3.umobilelab.local	
9	10.100.0.70	SEG-CAPR-4	Raspbian 8 (jessie) HypriotOS – Docker – AP	core	ssh://seg-capr-4.umobilelab.local	
10	10.4.1.4	SEG-CR-1	Raspbian 8 (jessie) HypriotOS – Docker – AP	core	ssh://seg-cr-1.umobilelab.local	
11	10.4.1.3	UNC1	- Service Controller	core	ssh://unc1.umobilelab.local	
12	10.2.1.4	A4BB-1	Android 4.2 – handheld device emulator	test	ssh://a4bb-1.umobilelab.local	
13	10.2.1.7	A4BB-2	Android 4.2 – handheld device emulator	test	ssh://a4bb-2.umobilelab.local	
14	10.2.1.10	A4BB-3	Android 4.2 – handheld device emulator	test	ssh://a4bb-3.umobilelab.local	
15	10.2.1.13	A4BB-4	Android 4.2 – handheld device emulator	test	ssh://a4bb-4.umobilelab.local	
16	10.2.1.5	A5BB-1	Android 5.1 – handheld device emulator	test	ssh://a5bb-1.umobilelab.local	http://a5bb-1.umobilelab.local:8000/
17	10.2.1.8	A5BB-2	Android 5.1 – handheld device emulator	test	ssh://a5bb-2.umobilelab.local	http://a5bb-2.umobilelab.local:8000/
18	10.2.1.11	A5BB-3	Android 5.1 – handheld device emulator	test	ssh://a5bb-3.umobilelab.local	http://a5bb-3.umobilelab.local:8000/
19	10.2.1.14	A5BB-4	Android 5.1 – handheld device emulator	test	ssh://a5bb-4.umobilelab.local	http://a5bb-4.umobilelab.local:8000/
20	10.2.1.3	LBB-1	Raspbian 8 (jessie) – handheld device emulator	test	ssh://lbb-1.umobilelab.local	
21	10.2.1.6	LBB-2	Raspbian 8 (jessie) – handheld device emulator	test	ssh://lbb-2.umobilelab.local	
22	10.2.1.9	LBB-3	Raspbian 8 (jessie) – handheld device emulator	test	ssh://lbb-3.umobilelab.local	
23	10.2.1.12	LBB-4	Raspbian 8	test	ssh://lbb-	

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 645124

			(jessie) – handheld device emulator		4.umobilelab.local	
24	10.2.1.2	ROBOT	Raspbian 8 (jessie) – handheld device emulator	test	ssh://robot.umobilelab.local	
25	10.1.1.6	APR-1	OpenWRT CHAOS CALMER (15.05.1, r48532) – AP	uan1	ssh://apr-1.umobilelab.local	http://apr-1.umobilelab.local
26	10.100.0.82	SEG-APR-1	Raspbian 8 (jessie) HypriotOS – Docker – AP	uan1	ssh://seg-apr-1.umobilelab.local	
27	10.1.1.7	APR-2	OpenWRT CHAOS CALMER (15.05.1, r48532) – AP	uan2	ssh://apr-2.umobilelab.local	http://apr-2.umobilelab.local
28	10.100.0.86	SEG-APR-2	Raspbian 8 (jessie) HypriotOS – Docker – AP	uan2	ssh://seg-apr-2.umobilelab.local	
29	10.1.1.8	APR-3	OpenWRT CHAOS CALMER (15.05.1, r48532) – AP	uan3	ssh://apr-3.umobilelab.local	http://apr-3.umobilelab.local
30	10.100.0.90	SEG-APR-3	Raspbian 8 (jessie) HypriotOS – Docker – AP	uan3	ssh://seg-apr-3.umobilelab.local	
31	10.1.1.9	APR-4	OpenWRT CHAOS CALMER (15.05.1, r48532) – AP	uan4	ssh://apr-4.umobilelab.local	http://apr-4.umobilelab.local
32	10.100.0.94	SEG-APR-4	Raspbian 8 (jessie) HypriotOS – Docker – AP	uan4	ssh://seg-apr-4.umobilelab.local	
33	10.4.1.1	MNVPN-1	VPN access	mgmt		http://lab1.umobile-project.eu/
34	10.4.1.5	NETMONITOR	Linux Centos – network management	mgmt		http://netmonitor.umobilelab.local/
35	10.4.1.6	REMOTEDESKTOP	windows 2003 sr2 – remote desktop helper	mgmt		

Table 3. UMOBILE Lab devices

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 645124

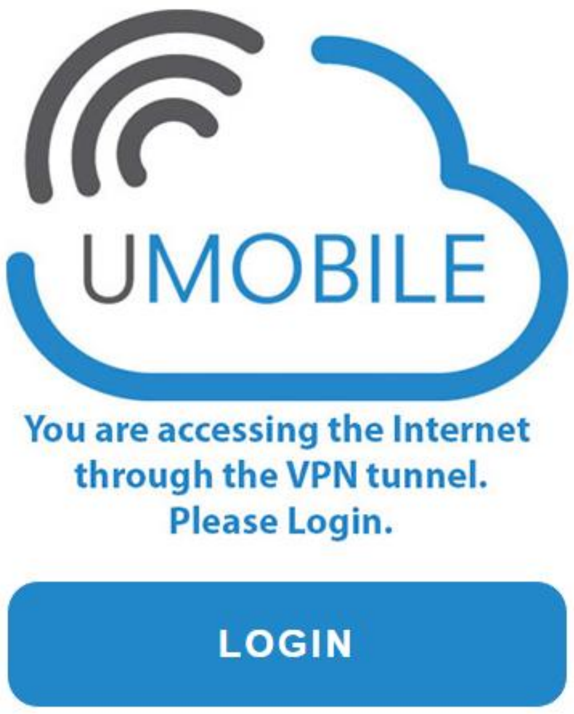

UMOBILE
You are accessing the Internet through the VPN tunnel. Please Login.
LOGIN

Figure 24. UMOBILE testbed - Captive portal

A1.2 How to install an Android APP

A5BB-x devices

A5BB-x devices use Android 5.1. Each device can be accessed via SSH/SCP by its IP.
To install a new App:

- download the desired App (.apk) from Internet;
- push the .apk into device (via an SCP client);
- execute via SSH the command "pm install app.apk" to install the App.

To use an App:

- enter the device via web (remote access).

A1.3 Testing NDN Network from Linux device

The LinuxBlackBox (LBB) are raspberry 3 device with Rasbian 8 OS. This device is equipped with two network connections

- first E1 with ethernet cable to TEST network to guarantee accessibility during the change of the wireless network
- second W1 with wireless 2.4 GHz to access on SSID of UMOBILE Network

It is possible to access the device using SSH on E1 IP address.

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 645124